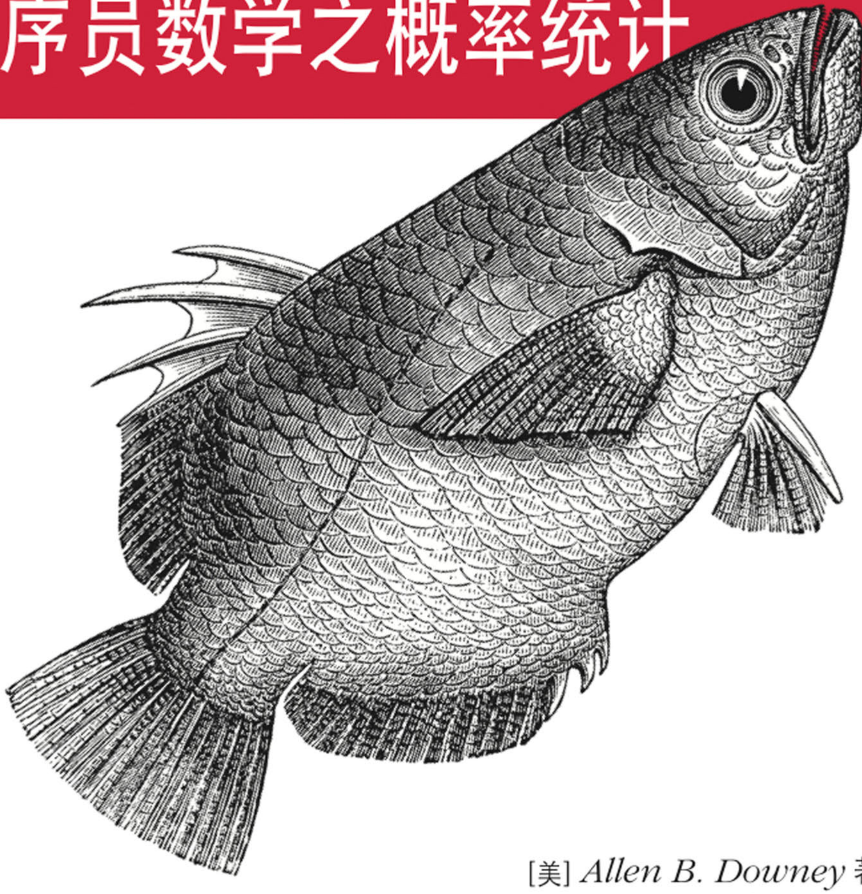
TURING 图灵程序设计丛书

Think Stats

统计思维

程序员数学之概率统计



[美] *Allen B. Downey* 著

张建锋 陈钢 译

O'REILLY®

人民邮电出版社
POSTS & TELECOM PRESS

数字版权声明

图灵程序设计丛书

统计思维

程序员数学之概率统计

Think Stats

[美] Allen B. Downey 著
张建锋　陈钢 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo
O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北　京

图书在版编目（C I P）数据

统计思维 : 程序员数学之概率统计 / (美) 唐尼
(Downey,A.B.) 著 ; 张建锋, 陈钢译. -- 北京 : 人民
邮电出版社, 2013.5
（图灵程序设计丛书）
书名原文: Think stats
ISBN 978-7-115-31737-7

Ⅰ. ①统… Ⅱ. ①唐… ②张… ③陈… Ⅲ. ①概率统
计 Ⅳ. ①O211

中国版本图书馆CIP数据核字(2013)第086989号

内 容 提 要

《统计思维：程序员数学之概率统计》是一本以全新视角讲解概率统计的入门图书。抛开经典的数学分析，Downey 手把手教你用编程理解统计学。概率、分布、假设检验、贝叶斯估计、相关性等，每个主题都充满趣味性，经编程解释后变得更为清晰易懂。

本书研究数据主要来源于美国全国家庭成长调查（NSFG）与行为风险因素监测系统（BRFSS），数据源及解决方案的相关代码全部开放，具体章节列出了大量学习和进阶资料，方便读者参考。

本书面向广大程序员和计算机专业的学生。

图灵程序设计丛书
统计思维：程序员数学之概率统计

◆ 著　[美] Allen B. Downey
　译　张建锋　陈　钢
　责任编辑　刘美英
　责任印制　焦志炜
◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街14号
　邮编　100061　电子邮件　315@ptpress.com.cn
　网址　http://www.ptpress.com.cn
　北京　印刷
◆ 开本：880×1230　1/32
　印张：5
　字数：125千字　2013年 5 月第 1 版
　印数：1－4 000册　2013年 5 月北京第 1次印刷
著作权合同登记号　图字：01-2012-8793号
ISBN 978-7-115-31737-7

定价：29.00元
读者服务热线：(010)51095186转604　印装质量热线：(010)67129223
反盗版热线：(010)67171154

版权声明

O’Reilly Media, Inc.介绍

O’Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O’Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O’Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O’Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O’Reilly 的会议和峰会聚集了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O’Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务或者面授课程，每一项 O’Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O’Reilly Radar 博客有口皆碑。”

——Wired

“O’Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O’Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O’Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

译者序

搞科研的都知道，论文发表情况是评判科研水平的主要指标，而学术期刊的影响因子则代表了期刊的档次，从而也就直接反映了论文的水平。在统计学领域，影响因子最高的期刊是什么？不是《皇家统计学会志，B 辑》（*JRSS B*），也不是大名鼎鼎的《统计年鉴》（*Annals of Statistics*）和《美国统计学会会刊》（*JASA*），而是《统计软件杂志》（*Journal of Statistical Software*）。什么？我推导了十几页公式，提出了如此高级的统计模型，证明了十几年无人搞定的定理，竟然还不如写两千行代码，发布一个统计分析的小工具？

这样的统计学世界让人感觉怪怪的，但现实如此。统计学的分析对象再也不放进打印在 A4 纸上的表格，靠普通计算器和教科书附录中的各种表格已经不可能实现现实所需的统计分析——离开了计算机，统计在现代科学和工程领域寸步难行。

不讲代码的统计学图书已经过时了！

你手中的这本书不是教你如何在 SPSS 中点鼠标，也不是如何在 SAS 中调用函数，而是介绍如何用 Python 这样的通用编程语言去获取和处理原始数据，去理解和实现统计分析，去跟上时代对统计的要求。

本书采用大量实例介绍了基本的统计分析知识，教你如何用 Python 编程语言实现各种分析，乃至画出统计图形。本书提供了绝对实用的配套

代码，你可以直接“拿来”对大量数据进行分析。作者 Allen Downey 是高校副教授，先后获得麻省理工学院计算机科学硕士和加州大学伯克利分校计算机科学博士学位。别惊讶，是的！计算机专业的教授写的统计学读本，这才是这本书最大的亮点。统计学和计算机在本书中紧紧地拥抱在一起，这是多美好的场景。

好吧，准备好键盘，让我们开始一次硅世界的统计学旅程！

张剑锋　陈钢

2013 年 3 月 26 日于深圳华大基因

致谢

感谢发起本书翻译的陈钢博士，他出色地引导我完成了本书部分章节的翻译工作。感谢图灵编辑部的刘美英女士，她对本书的校译和定稿提供了宝贵意见和辛勤劳动。感谢所有引导我走上统计学道路的人们，他们让我认识了一个如此精彩纷呈的世界。

感谢我的家人，你们是我在艰难中前行的不竭动力。

——张建锋

感谢我的合作者张建锋，他在关键时刻参与了本书的翻译工作。感谢图灵的傅志红、李松峰、刘美英等编辑的帮助和宽容，让我们能最终完成这本书的翻译工作。

谨以此书献给我的夫人王倩和儿子陈涤菲，你们是我所有努力的动力源泉。

——陈钢

目录

前言

为何写这本书

本书是一本全新的概率统计入门教材，重点介绍如何用统计学方法分析大型数据集。本书会介绍如何使用计算机实现各种统计方法，这有诸多优点。

- 学生可以通过编写程序来深化和检查自己对概念的理解。例如，编写计算最小二乘拟合、残差和判定系数的函数，编写和测试这些代码需要他们正确理解相关概念，消除各种可能的误解。
- 学生能够通过计算实验来验证统计学上的一些定理。例如，生成服从各种分布的样本来验证中心极限定理（Central Limit Theorem，CLT）。当发现服从帕累托分布的样本并没有收敛到正态分布时，他们肯定会记住中心极限定理的前提条件。
- 有些从数学上很难理解的概念可以很容易地用模拟方法来阐述。例如，通过蒙特卡罗模拟近似求出 p 值，就能说明 p 值的含义。
- 使用离散分布和计算方法可以把贝叶斯模拟之类在入门课程中很少见的内容讲清楚。例如，本书中有个练习要求学生计算“德国坦克问题”（German tank problem）的后验分布，通过理论分析很难得到答案，但用计算手段却很容易得出结果。

- 因为学生使用的是通用编程语言 Python，所以他们可以导入各种来源的数据，并不局限于那些已经为特定统计工具整理好的数据。

本书内容是按项目来组织的。在我的课上，学生会用一个学期的时间来做一个项目，在此期间他们要提出统计学问题、找到合适的数据集，并用学到的技术来处理这些数据。

为了演示我想要学生做的分析，本书会有一个贯穿全书的实例。它所使用的数据主要有以下两个来源。

- 由美国疾病控制与预防中心（Centers for Disease Control and Prevention，CDC）主持的全国家庭成长调查（National Survey of Family Growth，NSFG）项目,该项目是为了收集美国人的“家庭生活、婚姻状况、生育、避孕和男女健康等信息”（参见 http://cdc.gov/nchs/nsfg.htm ）。
- 由全国慢性病预防和健康促进中心主导的行为风险因素监测系统（the Behavioral Risk Factor Surveillance System，BRFSS），该系统旨在跟踪“美国人的健康状况和危险行为”（参见 http://cdc.gov/BRFSS/）。

其他例子所使用的数据则来自 IRS（美国国税局）、美国人口普查和波士顿马拉松比赛。

本书的写作原则

要写一本新教材，通常的做法是参考一堆老教材。很多书的内容因此都会重复，且行文组织也大同小异，出现了不少以讹传讹的情况。史蒂芬 • 古尔德在他的文章中就提到过一个关于猎狐犬的例子[1]。但我可没有这样做，我在撰写本书过程中几乎没有参考过任何已出版的材料，主要原因如下。

- 本书的目的是要探索这方面的新方法，所以对已有的方法我并不想讲太多。

注 1：Stephen Jay Gould（1941—2002），美国著名的进化生物学家、古生物学家、科学史学家和科普作家；很多教科书中说始祖马跟小猎狐犬差不多大，史蒂芬 • 古尔德在他的文章中纠正了这一错误观点（译者注），狗大概只有始祖马的一半大（见 http://wikipedia.org/wiki/Hyracotherium）。

- 我想让所有人都可以自由地使用本书内容，因此要确保书中所有内容都没有版权限制。
- 很多读者都不方便到图书馆借阅纸版书，所以本书尽量只引用能在网上免费查阅的内容。
- 传统媒体的支持者认为只使用网络资源是一种偷懒的行为，而且内容也不可靠。我认同前半句话，但不同意后半句，所以我要亲自证明。

我用得最多的资源是维基百科，其内容涉及方方面面，每个人都可以查阅。总体看来，我所读过的关于统计学的文章整体质量都不错（不过我也做了一些修改）。本书引用了不少维基百科的链接，建议读者去读一读这些网页内容。这些页面很多时候都是对本书所介绍内容的很好补充和完善。除非有充分的理由修改，否则本书中所使用的术语和概念与维基百科基本保持一致。

其他实用资源包括 Wolfram MathWorld（http://mathworld.wolfram.com/），当然还有 Google。我还用到了两本书，David MacKay 的 *Information Theory, Inference, and Learning Algorithms*[2] 和 Press 等人合著的 *Numerical Recipes in C*，前者是我学习贝叶斯统计的书。不过，因为这两本书都可以在网上找到，所以并不违反我之前所说的原则。

贡献者名单

对本书有任何意见或者建议，请发送邮件到 downey@allendowney.com。如果本书根据你的反馈进行了修改，我会把你加入贡献者名单（除非你主动要求不加入）。

最好能写清具体出错的句子，以便我找出问题。只给出页码和章节号也行，但不方便。非常感谢！

- Lisa Downey 和 June Downey 阅读了早期的书稿，提出了很多意见和建议。

注 2：中译本《信息论、推理与学习算法（翻译版）》由高等教育出版社出版（2006）；《C 语言数值算法程序大全（第二版）》由电子工业出版社出版（1995）。——编者注

- Steven Zhang 发现了一些错误。
- Andy Pethan 和 Molly Farison 帮我调试了一些解题程序，Molly 指出了一些拼写错误。
- Andrew Heine 发现了 `error` 函数中的错误。
- Nikolas Akerblom 博士告诉了我始祖马的大小。
- Alex Marrow 改进了一个代码示例的说明。
- Jonathan Street 在关键时刻发现了一个错误。
- Gábor Lipták 发现了一个拼写错误和接力赛问题的答案。
- 感谢 Kevin Smith 和 Tim Arnold 写的 plasTeX，我用它将此书转成了 DocBook。
- George Caplan 的建议让本书更加清晰易懂。

排版约定

本书使用的排版约定如下。

- 楷体
  表示新的术语。
- 等宽字体
  表示程序片段，也用于在正文中表示程序中使用的变量、函数名、命令行代码、环境变量、语句和关键词等代码文本。
- 加粗的等宽字体
  表示应该由用户逐字输入的命令或者其他文本。
- 倾斜的等宽字体
  表示应该由用户输入的值或根据上下文决定的值替换的文本。

这个图标代表小窍门、建议或说明。

这个图标代表警告信息。

示例代码

本书就是要帮读者解决实际问题的。也许你需要在自己的程序或文档中用到本书中的代码。除非大段大段地使用，否则不必与我们联系取得授权。因此，用本书中的几段代码写成一个程序不用向我们申请许可。但是销售或者分发 O'Reilly 图书随附的代码光盘则必须事先获得授权。引用书中的代码来回答问题也无需我们授权。将大段的示例代码整合到你自己的产品文档中则必须经过许可。

使用我们的代码时，希望你能标明它的出处。出处一般要包含书名、作者、出版商和 ISBN，例如：*Think Stats* by Allen B. Downey (O'Reilly). Copyright 2011 Allen B. Downey, 978-1-449-30711-0。

如果还有其他使用代码的情形需要与我们沟通，可以随时与我们联系：permissions@oreilly.com。

Safari® Books Online

Safari Books Online（www.safaribooksonline.com）是应需而变的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。

Safari Books Online 是技术专家、软件开发人员、Web 设计师、商务人士和创意人士开展调研、解决问题、学习和认证培训的第一手资料。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、

New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

http://www.oreilly.com/catalog/0636920020745

中文版地址：

http://www.oreilly.com.cn/index.php?func=book&isbn=978-7-115-31737-7

对于本书的评论和技术性问题，请发送电子邮件到：

bookquestions@oreilly.com

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

http://www.oreilly.com

我们在 Facebook 的地址如下：

http://facebook.com/oreilly

请关注我们的 Twitter 动态：

http://twitter.com/oreillymedia

我们的 YouTube 视频地址如下：

http://www.youtube.com/oreillymedia

第 1 章

程序员的统计思维

本书讨论如何将数据转换为知识。数据是廉价的（至少相对而言如此），但知识却异常宝贵。

我会介绍以下三门相互关联的学科。

- 概率论

  主要研究随机事件。人们对某些事件发生的可能性高低一般都有直观的认识，所以未经特殊训练就会使用“可能”、“不可能”之类的词汇。但本书会介绍如何量化这种可能性。

- 统计学

  统计学旨在根据数据样本推测总情况。大部分统计分析都基于概率，所以这两方面的内容通常兼而有之。

- 计算

  量化分析的最佳工具。计算机是处理统计量的常用工具。此外，计算实验还有助于理解概率论和统计学中的概念。

本书的主要目的就是要让懂编程的人通过编程来理解概率论和统计学。人们通常是从数学角度讲解概率论和统计学，而且很多人也因此学会了概率论和统计学。但在概率论和统计学中，有很多概念从

数学角度很难理解，但如果用计算方法就比较容易。

记得我妻子怀上我们第一个孩子时，我听到过这样一个问题：第一胎多在预产期后出生吗？本章接下来介绍的例子就源自这个问题。

1.1 第一个孩子出生晚吗

如果在 Google 上搜索这个问题，你会看到大量的相关讨论。有些人说确实如此，也有人说这没根据，还有人持完全相反的观点：第一个孩子会在预产期之前出生。

在这类讨论中，人们会用各种数据来证明自己的说法，常见的例子如下。

> “我有两个朋友最近都刚生了第一个孩子，两个宝宝的出生时间都比预产期晚了差不多两周。”

> “我的第一个孩子晚了两周才出生，我想我的第二个孩子会提前两周。”

> “我觉得这没道理，因为我姐姐是我妈妈的第一个孩子，她就提前出生了，我的几个表姐也一样。”

诸如此类的传闻称为经验之谈（anecdotal evidence），因为它们基于非公开发表的数据，而且通常是个人感受。在非正式场合，这类说辞没问题，所以这里并不是说上述观点不对。问题在于，我们需要更有说服力的证据和更可靠的结论。但这些经验之谈显然做不到这一点，原因如下。

- 观察的数量太少

  第一胎婴儿的妊娠期比较长，但这种差异可能在自然波动范围内。这种情况下，我们需要比较大量孕妇的妊娠期数据才能判断这种差异是否真的存在。

- 选择偏差

  第一胎婴儿出生比较晚的父母会更有兴趣加入这样的讨论。这种对数据进行选择的过程就会导致结果不准确。

- 确认偏差

  相信这种说法的人会提供支持示例，而怀疑这种说法的人则会引用反例。

- 不准确

  传闻通常都是个人的经历，在记忆、表述和复述等方面都会不准确。

那么，更好的做法是什么呢？

1.2 统计方法

为了解决上述经验之谈的种种不足，我们会运用以下统计学手段。

- 收集数据

  使用大型全国性调查的数据，这些数据是为得出美国人口方面可靠的统计推断而专门收集的。

- 描述性统计

  计算能总结数据的统计量，并评测各种数据可视化的方法。

- 探索性数据分析

  寻找模式、差异和其他能解答我们问题的特征。同时，我们会检查不一致性，并确认其局限性。

- 假设检验

  在发现明显的影响时（比如两个族群间的差异），我们需要评判这种影响是否真实，也就是说是否是因为随机因素造成的。

- 估计

  我们会用样本数据推断全部人口的特征。

通过这些步骤，绕过各种陷阱，我们就能得到更加合理也更可能正确的结论。

1.3 全国家庭成长调查

美国疾病控制与预防中心（CDC）从 1973 年开始推行全国家庭成长调查（NSFG），目的是收集（美国）“家庭的生活、婚姻状况、生育、避孕和男女健康信息。调查的结果用于……制定健康服务和健康教育计划，以及对家庭、生育和健康的统计研究”。[1]

我们会利用调查收集的数据来研究诸如“第一个小孩是否出生得较晚”之类的问题。为了有效使用这些数据，我们需要理解这个调查是怎么设计的。

NSFG 是一个*横断面研究*（cross-sectional study），意思就是它的数据是一群人在某个时间点的情况。另一种常见方法是*纵贯研究*（longitudinal study），就是在一段时间内反复观察同一群人。

NSFG 已经进行了 7 次，每次称为一个*周期*（cycle）。我们会使用来自 Cycle 6 的数据，这些数据是在 2002 年 1 月到 2003 年 3 月间收集的。

NSFG 的目的是得到关于*人口情况*的一些结论，调查对象是 15 到 44 岁的美国人。

参与调查的人称为*被调查者*（respondent），一组被调查者就称为*队列*（cohort）。通常，横断面研究意味着具有*代表性*，即目标人群中的每一个人都有同等的几率参与调查。当然，实际很难实现这种理想状况，但执行调查的人会尽可能地做到这一点。

NSFG 不具有代表性，而是有意进行了*过采样*（oversample）。设计者所调查的西班牙裔、非裔美国人和青少年的比例都高于他们在美国人口中的比例。过采样这些人群是为了确保其中的被调查者数量够大，从而得到有效的统计推断。

当然，过采样增大了根据调查结果推断全体人口结论的难度。稍候我们会继续讨论这一点。

注 1：参见 http://cdc.gov/nchs/nsfg.htm。

习题1-1

尽管 NSFG 已经进行了 7 次，但它并不是纵贯研究。阅读维基百科页面 http://wikipedia.org/wiki/Cross-sectional_study 和 http://wikipedia.org/wiki/Longitudinal_study 可以弄清楚原因。

习题1-2

这个练习需要从 NSFG 下载数据，本书接下来会用到这些数据。

1. 打开 http://thinkstats.com/nsfg.html，阅读数据的使用协议，然后点击“I accept these terms”（假设你确实同意）。

2. 下载 2002FemResp.dat.gz 和 2002FemPreg.dat.gz 两个文件。前者是被调查者文件，每一行代表一个被调查者，总共 7643 个女性被调查者。后者是各个被调查者的怀孕情况。

3. 调查的在线资料地址：http://www.icpsr.umich.edu/nsfg6 。浏览左侧导航栏中调查的各部分，大致了解一下其中的内容。还可以在 http://cdc.gov/nchs/data/nsfg/nsfg_2002_questionnaires.htm 上阅读调查问卷的内容。

4. 本书的配套网站提供了处理 NSFG 数据文件的代码。从 http://thinkstats.com/survey.py 下载，然后在放置数据文件的目录中运行。程序会读取数据文件，然后会显示每个文件的行数：

```
Number of respondents 7643
Number of pregnancies 13593
```

5. 浏览一下代码，大致了解一下其功能。下一节会详细介绍。

1.4 表和记录

诗人、哲学家 Steve Martin 曾说：

> “Oeuf”就是egg，“chapeau”就是hat。好像所有的东西在法语中都跟在英语中的叫法不一样。

跟法语一样，数据库程序员的语言也跟我们的日常语言稍有不同。因为我们要谈到数据库，所以有必要学习一些专业术语。

被调查者文件中的每一行都表示一个被调查者。这行信息称为一条*记录*（record），组成记录的变量称为*字段*（field），若干记录的集合就组成了一个*表*（table）。

看一下 survey.py 中的代码，就会看到 `Record` 和 `Table` 这两个类的定义，前者是代表记录的对象，后者则是表示表的对象。

`Record` 有两个子类，分别是 `Respondent` 和 `Pregnancy`，两者分别是被调查者和怀孕的记录。目前这些类暂时还是空的，其中还没有用于初始化其属性的 `init` 方法。我们会用 `Table.MakeRecord` 方法将一行文本转换成一个 `Record` 对象。

`Table` 也有两个子类 `Respondents` 和 `Pregnancies`。这两个类的 `init` 方法设置了数据文件的默认名称和要创建的记录的类型。每个 `Table` 对象都有一个 `records` 属性，是一个 `Record` 对象的列表。

每个 `Table` 的 `GetFields` 方法返回一个指定记录字段的元组（tuple）列表，这些字段就是 `Record` 对象的属性。

例如，下面是 `Pregnancies.GetFields`：

```
def GetFields(self):
        return [
            ('caseid', 1, 12, int),
            ('prglength', 275, 276, int),
            ('outcome', 277, 277, int),
            ('birthord', 278, 279, int),
            ('finalwgt', 423, 440, float),
            ]
```

第一个元组的意思从第 1 列到第 12 列是 `caseid` 字段，且类型为整数。每个元组包含如下信息。

- `field`
  保存该字段的属性的名称。大部分情况下，我使用 NSFG 编码手册中的名称，全部用小写。

- `start`
  该字段的起始列编号。例如，`caseid` 的起始编号是 1。可以在 NSFG 编码手册中查询这些编号：http://www.icpsr.umich.edu/nsfg6/。

- `end`
  该字段的结束列编号。例如，`caseid` 的结束列编号是 12。跟 Python 中不一样，这里的结束列*也是*该字段的一部分。

- 转换函数
  将字符串转换成其他类型的函数。可以用内置的函数，比如 `int` 和 `float`，也可以使用用户自定义的函数。如果转换失败，属性的值就会是字符串 `'NA'`。如果某个字段不需要转换，可以使用 `identity` 函数或是 `str` 函数。

从 `pregnancy` 记录中可以得到以下变量。

- `caseid`
  被调查者的整数 ID。

- `prglength`
  怀孕周期，单位是周。

- `outcome`
  怀孕结果的整数代码。代码 1 表示活婴。

- `birthord`
  正常出生的婴儿的顺序。例如，第一胎婴儿的编号是 1。如果没有正常出生，该字段为空。

- `finalwgt`
  被调查者的统计权重。这是一个浮点值，表示这名被调查者所代表的人群在美国总人口中的比例。过采样人群的权重偏低。

如果你仔细阅读编码手册，就会发现这些变量大部分都经过了*重编码*（recode），也就是说这并不是调查所采集的*原始数据*，而是根据原始数据计算出来的。

例如，第一胎活婴的 prglength 在原始数据中有变量 wksgest（妊娠周数）时就等于该变量的值，否则就会用 mosgest * 4.33（妊娠月数乘以每个月的平均周数）估计出来。

重编码通常遵循数据一致性和准确性原则。除非有特别原因一定要使用原始数据，否则就应该直接使用重编码后的数据。

你可能还发现了 Pregnancies 有 Recode 方法，用来做一些其他的检查和重编码工作。

习题1-3

在这个练习中，我们会编写一个程序来看看 Pregnancies 表中的数据。

1. 在 survey.py 和数据文件的目录中创建一个 first.py 文件，然后将下面的代码输入或复制到文件中：

   ```
   import survey
   table = survey.Pregnancies()
   table.ReadRecords()
   print 'Number of pregnancies', len(table.records)
   ```

   结果应该是 13 593 条怀孕记录。

2. 编写一个循环遍历表（table），计算其中活婴的数量。查阅临床结果（outcome）的文档，确认你的结果跟文档中的总结一致。

3. 修改这个循环，将活婴的记录分成两组：一组是第一胎出生；另一组是其他情况。再看一些出生顺序（birthord）的文档，看看你的结果跟文档中的结果是否一致。

   在处理新的数据集时，这种检查对于发现数据中的错误和不一致性、检查程序中的错误以及检验对字段编码方式的理解是否正确等都是很有用的。

4. 分别计算第一胎婴儿和其他婴儿的平均怀孕周期（单位是周）。两组之间有差异吗？差异有多大？

从 http://thinkstats.com/first.py 可下载这个练习的答案。

1.5 显著性

在前面的练习中，我们比较了第一胎婴儿和其他婴儿的妊娠期。如果一切顺利，读者会发现第一胎婴儿的出生时间比其他婴儿的出生时间平均晚 13 个小时。

类似这样的差异称为*直观效应*（apparent effect），意思就是似乎发生了有意思的事情，但还不确定。我们还需要考虑以下问题。

- 如果两组的均值不一样，其他汇总统计量如何，比如中位数和方差？我们能更精确地描述它们之间的差异吗？
- 有没有可能这两组实际上是一样的，而我们所观察到的这种差异只是随机产生的？如果是，那这个结论就不是*统计显著*的。
- 这种直观效应有没有可能是因为选择偏差或是实验设置中的错误导致的？如果是，那么这种直观效应就是*人为的*，也就是我们意外创造的，而并非发现了事实。

本书接下来的大部分内容都是为了回答这些问题。

习题1-4

学习统计学的最好方法就是从一个自己感兴趣的项目开始。有没有“第一胎婴儿出生较晚”这类吸引你的问题来研究？

思考自己感兴趣的问题，例如传统观念、有争议的话题或是有社会影响的问题，看看你能否将这些问题转换成统计学问题。

寻找能解决该问题的数据。国外政府是很好的数据来源，因为公共研究的数据通常都是免费的[2]。另一个查找数据的好去处是 Wolfram Alpha，其中收集了很多经过验证的高质量的数据集，网址是 http://wolframalpha.com。Wolfram Alpha 的搜索结果是有版权限制的，在使用之前应该阅读一下协议。

注 2：在撰写这段内容的时候，英国某法院规定“信息自由法案”（Freedom of Information Act）也适用于科学研究数据。

Google 和其他的一些搜索引擎也能帮你寻找数据，但网络上各种资源的质量高低不一，判断起来不容易。

如果发现已经有人回答了你的问题，要仔细看看回答是否合理。数据和分析中的缺陷可能会导致结论不可靠。如果是这样，你应该采用不同的方法来分析数据，或者是寻找其他更好的数据来源。

如果已发表的论文回答了你的问题，那就应该能弄到原始数据，很多作者都会在网上提供。但如果数据涉及个人隐私，最好联系一下作者，告诉他你要如何使用数据，或是接受特定的使用协议。坚持到底！

1.6 术语

- 经验之谈（anecdotal evidence）
  个人随意收集的证据，而不是通过精心设计并经过研究得到的。
- 直观效应（apparent effect）
  表示发生了某种有意思的事情的度量或汇总统计量。
- 人为（artifact）
  由于偏差、测量错误或其他错误导致的直观效应。
- 队列（cohort）
  一组被调查者。
- 横断面研究（cross-sectional study）
  收集群体在特定时间点的数据的研究。
- 字段（field）
  数据库中组成记录的变量名称。
- 纵贯研究（longitudinal study）
  跟踪群体，随着时间推移对同一组人反复采集数据的研究。
- 过采样（oversampling）
  为了避免样本量过少，而增加某个子群体代表的数量。

- 总体（population）
  要研究的一组事物，通常是一群人，但这个术语也可用于动物、蔬菜和矿产。
- 原始数据（raw data）
  未经或只经过很少的检查、计算或解读而采集和重编码的值。
- 重编码（recode）
  通过对原始数据进行计算或是其他逻辑处理得到的值。
- 记录（record）
  数据库中关于一个人或其他对象的信息的集合。
- 代表性（representative）
  如果人群中的每个成员都有同等的机会进入样本，那么这个样本就具有代表性。
- 被调查者（respondent）
  参与调查的人。
- 样本（sample）
  总体的一个子集，用于收集数据。
- 统计显著（statistically significant）
  若一个直观效应不太可能是由随机因素引起的，就是统计显著的。
- 汇总统计量（summary statistic）
  通过计算将一个数据集归结到一个数字（或者是少量的几个数字），而这个数字能表示数据的某些特点。
- 表（table）
  数据库中若干记录的集合。

第 2 章

描述性统计量

2.1 均值和平均值

前一章提到了三个汇总统计量，均值、方差和中位数，但没有解释它们的具体含义。在介绍其他内容之前，我们先来看看这三个统计量。

如果有一个包含 n 个值的样本 x_i，那么它们的均值 μ 就等于这些值的总和除以值的数量，即：

$$\mu = \frac{1}{n}\sum_i x_i$$

均值（mean）和平均值（average）在很多情况下可以不加区分地使用，但这里还要强调一下两者的区别：

- 样本的“均值”是根据上述公式计算出来的一个汇总统计量；
- “平均值”是若干种可以用于描述样本的典型值或集中趋势（central tendency）的汇总统计量之一。

有时候均值可以很好地描述一组值。例如，苹果大小都差不多（至少在超市里出售的苹果都是如此），因此，如果我买了 6 个苹果，总重量是 3 磅，那么就可以说每个苹果的重量大概是半磅。

但是南瓜就不一样了。假如我在花园里种了一些蔬菜，到了收获的时候，我收获了三个装饰用的南瓜，每个 1 磅重；两个制南瓜饼的南瓜，每个 3 磅重；还有一个重达 591 磅的大西洋巨型南瓜。这组样本的均值是 100 磅，但如果我告诉你："我种的南瓜的平均重量是 100 磅。"那就有问题了，至少这也是一种误导。

在这个例子中，平均值是没有意义的，因为"典型"的南瓜是不存在的。

2.2 方差

既然一个值不能概括南瓜的重量，那两个值应该会好一些：均值和*方差*。

均值是为了描述集中趋势，而方差则是描述*分散*情况。一组值的方差等于：

$$\sigma^2 = \frac{1}{n}\sum_i (x_i - \mu)^2$$

其中 $x_i - \mu$ 叫做*离均差*（deviation from the mean），因此方差为该偏差的方均值，这也是用 σ^2 表示的原因。方差的平方根 σ 叫做*标准差*。

方差本身不太好解释，其中一个问题就是它的单位很奇怪。在南瓜重量的例子里，度量单位是磅，所以方差的单位就是磅的平方。标准差的含义就明确多了，在上例中单位为磅。

习题2-1

先从 http://thinkstats.com/thinkstats.py 下载脚本文件，其中的函数在整本书中都会有用。这些函数的文档可以参考这里 http://thinkstats.com/thinkstats.html。

编写一个 `Pumpkin` 函数，并调用 thinkstats.py 中的函数计算上一节中南瓜重量的均值、方差和标准差。

习题2-2

重用 survey.py 和 first.py 中的代码计算第一胎婴儿的怀孕周期和其他婴儿的怀孕周期的标准差。这两组数据的分散情况一样吗？

跟标准差的差异相比，均值的差异有多大？对于该差异的统计显著性，这个比较说明了什么？

如果以前接触过方差，你会发现方差计算公式中的除数是 $n-1$，而不是 n，这个统计量叫做“样本方差”，用于通过样本估计总体的方差。第 8 章我们会再次介绍这个概念。

2.3 分布

汇总统计量简单明了，但风险也大，因为它们很有可能会掩盖数据的真相。另一种方法就是看数据的*分布*（distribution），它描述了各个值出现的频繁程度。

表示分布最常用的方法是*直方图*（histogram），这种图用于展示各个值出现的频数或概率。

在这里，*频数*指的是数据集中一个值出现的次数，跟声音的音高和无线电信号的调频没有关系。*概率*就是频数除以样本数量 n。

在 Python 中，计算频数最简单的方法就是用字典。给定一个序列 `t`：

```
hist = {}
for x in t:
    hist[x] = hist.get(x, 0) + 1
```

得到的结果是一个将值映射到其频数的字典。将其除以 n 即可把频数转换成概率，这称为*归一化*（normalization）：

```
n = float(len(t))
pmf = {}
for x, freq in hist.items():
    pmf[x] = freq / n
```

归一化之后的直方图称为 PMF（Probability Mass Function，概率质量

函数），这个函数是值到其概率的映射（习题 6-5 中会介绍“质量”的含义）。

将 Python 中的字典称为函数可能会让部分读者感到困惑。在数学中，函数就是一组值到另一组值的映射。在 Python 中，我们通常用函数对象表示数学中的函数，但这个例子中用的是字典（字典也被称为“映射”，所以称其为“函数”也是可以理解的）。

2.4 直方图的表示

我编写了一个名叫 Pmf.py 的 Python 模块，其中定义了用于表示直方图的 `Hist` 对象，以及表示 PMF 的 `Pmf` 对象。可从 http://thinkstats.com/Pmf.html 阅读它的文档，从 http://thinkstats.com/Pmf.py 下载源代码。

`MakeHistFromList` 函数接受一个序列，并返回一个新的 `Hist` 对象。可以在 Python 的交互模式下测试一下：

```
>>> import Pmf
>>> hist = Pmf.MakeHistFromList([1, 2, 2, 3, 5])
>>> print hist
<Pmf.Hist object at 0xb76cf68c>
```

`Pmf.Hist` 的意思是这个对象属于 `Pmf` 模块中定义的 `Hist` 类。一般情况下，书中的类和函数名首字母大写，变量名首字母小写。

`Hist` 对象提供了查找值及其概率的方法。`Freq` 方法接收一个值，并返回它的频数：

```
>>> hist.Freq(2)
2
```

如果所查找的值不存在，那么频数就是 0。

```
>>> hist.Freq(4)
0
```

`Values` 方法会返回未经排序的 `Hist` 类的对象的所有值：

```
>>> hist.Values()
[1, 5, 3, 2]
```

要按序遍历这些值，可以用内置的 sorted 函数：

```
for val in sorted(hist.Values()):
    print val, hist.Freq(val)
```

如果要查找所有的频数，用 Items 会更高效。它会返回一组未经排序的值—频数对：

```
for val, freq in hist.Items():
     print val, freq
```

习题2-3

一个分布的众数就是它的最频繁值（见 http://wikipedia.org/wiki/Mode_(statistics)）。编写一个 Mode 函数，以 Hist 对象为参数，返回最频繁值。

再来一个更有挑战的，编写一个 AllModes 函数，参数还是 Hist 对象，但返回的是按频数降序排列的值—频数对。提示：operator 模块中有个 itemgetter 函数可以按键值排序。

2.5 绘制直方图

Python 中有不少画图的包。这里我们要演示的是 pyplot，来自 matplotlib（http://matplotlib.sourceforge.net）。

很多 Python 安装程序中都带有这个包。启动 Python 解释器，输入以下命令就可以查看是否安装了这个包：

```
import matplotlib.pyplot as pyplot
pyplot.pie([1,2,3])
pyplot.show()
```

如果安装了 matplotlib，就应该能看到一个饼图，否则就说明还没有安装。

直方图和概率质量函数通常画成条状图。pyplot 中绘制条状图的函数是 bar。Hist 对象中有一个 Render 方法，会返回一个排序后的值

列表，以及相应的频数，下面是 `bar` 函数所需的参数：

```
>>> vals, freqs = hist.Render()
>>> rectangles = pyplot.bar(vals, freqs)
>>> pyplot.show()
```

我编写了一个 myplot.py，提供了绘制直方图的函数、概率质量函数，以及你将要看到的其他对象。它的文档在 thinkstats.com/myplot.html 上，从 thinkstats.com/myplot.py 可以下载到代码。或者你喜欢的话，也可以直接用 `pyplot`，可以在网上找到它的文档。

图 2-1 是第一胎婴儿和非第一胎婴儿怀孕周期直方图。

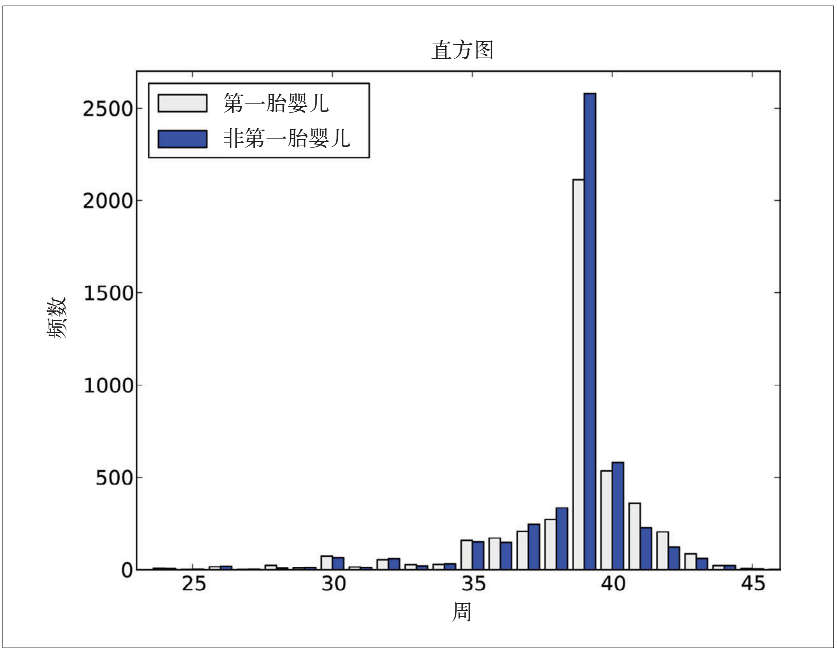


图 2-1：怀孕周期直方图

直方图很有用，因为它可以非常直观地展现数据的以下特征。

- 众数

  分布中出现次数最多的值叫做众数。在图 2-1 中，众数显然是在 39

周，众数是最适合描述典型值的汇总统计量。

- 形状

  以众数为中心，整个分布是不对称的；在右侧下降很快，而在左侧下降比较慢。从医学的角度来看，这是对的。婴儿经常会提前来到这个世界，但却很少有呆在妈妈肚子里超过 42 周的。此外，分布的右侧到了 42 周就被截断了，因为到了这个时候医生会采取必要的措施。

- 异常值

  远离众数的值叫做异常值（outlier）。其中有些只是罕见情况，比如 30 周出生的婴儿。但有些很有可能是汇总或者记录数据的某个环节中的失误导致的。

直方图直观地展示了数据的一些特征，但通常用来比较两个分布意义不大。在这个例子中，“第一胎婴儿”的数量要比“非第一胎婴儿”的数量少。因此，直方图中某些明显差异是由样本数量造成的。可以用 PMF 来解决这个问题。

2.6 表示概率质量函数

Pmf.py 中定义了用于表示概率质量函数的 `Pmf` 类。这里稍微解释一下：`Pmf` 是模块的名称，也是类的名称，因此这个类的全称是 `Pmf.Pmf`。`pmf` 则通常被我用做变量名。在正文中，我用 PMF 指代通常意义上的概率质量函数，这个简写与我的实现无关。

用 `MakePmfFromList` 方法创建 `Pmf` 对象，其参数是一组值：

```
>>> import Pmf
>>> pmf = Pmf.MakePmfFromList([1, 2, 2, 3, 5])
>>> print pmf
<Pmf.Pmf object at 0xb76cf68c>
```

`Pmf` 对象跟 `Hist` 对象有很多类似的地方，两者的 `Values` 方法和 `Items` 方法是一样的。最大区别在于 `Hist` 是将值映射到一个用整数表示的数量，而 `Pmf` 是将值映射到一个用浮点数表示的概率。

用 Prob 查看给定值的概率：

```
>>> pmf.Prob(2)
0.4
```

可以通过增加某个值的概率来修改现有的 Pmf：

```
>>> pmf.Incr(2, 0.2)
>>> pmf.Prob(2)
0.6
```

还可以将概率扩大若干倍：

```
>>> pmf.Mult(2, 0.5)
>>> pmf.Prob(2)
0.3
```

如果修改 Pmf，有可能导致整个 PMF 不再是归一化的，也就是说所有概率的总和不再等于 1。可以用 Total 方法检查一下，该方法会返回所有概率的总和：

```
>>> pmf.Total()
0.9
```

要重新归一化，调用 Normalize：

```
>>> pmf.Normalize()
>>> pmf.Total()
1.0
```

Pmf 对象提供的 Copy 方法可以复制 Pmf 对象，修改复制出来的 Pmf 对象不会影响原来的数据。

习题2-4

根据维基百科："生存分析是统计学的一个分支，涉及生物体的死亡和机械系统故障。"详见 http://wikipedia.org/wiki/Survival_analysis。

作为生存分析的一部分，计算剩余使用寿命（例如某个机械部件还能用多长时间）是大有用处的。如果知道使用寿命的分布和部件的使用时间，就可以计算出剩余使用寿命的分布。

编写一个函数 `RemainingLifetime`，参数是表示使用寿命的 `Pmf` 对象和使用时间，返回一个表示剩余使用寿命分布的 `Pmf` 对象。

习题2-5

2.1 节介绍过，通过累加各个元素并除以 n 可以算出样本的均值。对于给定的 PMF，也可以算出均值，但计算过程略有不同：

$$\mu = \sum_i p_i x_i$$

其中 x_i 是 PMF 中的值，p_i=PMF(x_i)。同样，也可以计算方差：

$$\sigma^2 = \sum_i p_i (x_i - \mu)^2$$

编写两个函数，`PmfMean` 和 `PmfVar`，两者的参数都是一个 `Pmf` 对象，分别计算它的均值和方差。看一看结果是否跟 Pmf.py 中的 `Mean` 和 `Var` 方法的结果一致。

2.7 绘制概率质量函数

绘制 Pmf 的常用方法有以下两种。

- 采用柱状图，可以用 pyplot.bar 或 myplot.Hist。如果 Pmf 中的值不多，柱状图就比较合适。
- 采用折线图，可以用 pyplot.plot 或者 myplot.Pmf。如果 Pmf 中的值较多，且比较平滑，折线图就比较合适。

图 2-2 用柱状图展示了怀孕周期的 PMF。借助 PMF，我们可以更清晰地看出分布的差异。第一胎婴儿似乎较少按时出生（39 周），而倾向于比这个时间晚一些（41 周或 42 周）。

生成本章图片的代码可以从 http://thinkstats.com/descriptive.py 下载。运行代码需要安装其所需的模块和 NSFG 的数据（参见 1.3 节）。

注意：`pyplot` 中的 `hist` 函数接受一个序列，然后计算其直方图，并画出来。因为我用的是 `Hist` 对象，所以通常不使用 `pyplot.hist`。

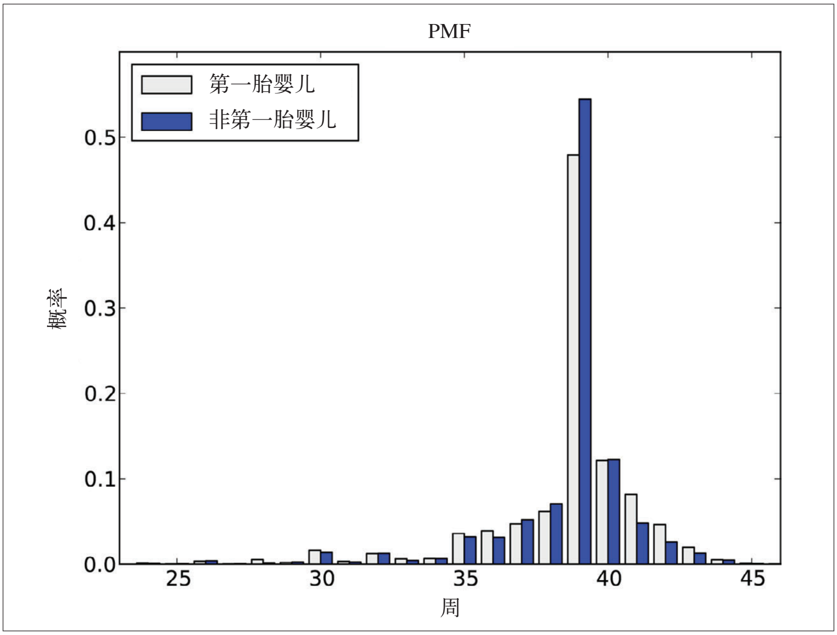


图 2-2：怀孕周期的 PMF

2.8 异常值

异常值就是远离集中趋势的值。异常值有可能是采集和处理数据过程中的错误导致的，也有可能是罕见的正确结果。有时对这些异常值修剪（trim）既恰当又十分有用。

在活婴的怀孕周期数据中，最低的十个值是 {0, 4, 9, 13, 17, 17, 18, 19, 20, 21}。低于 20 周的值肯定是错误的，只有高于 30 周的值正确的可能性才比较大。介于两者之间的值就很难解释了。

另一方面，最大的几个值分别是：

```
weeks  count
43     148
44      46
45      10
46       1
```

```
47        1
48        7
50        2
```

强调一下，有些值很有可能是错误的，但不好说。一种处理方法是对一定比例的最高和最低值修剪（参见 http://wikipedia.org/wiki/Truncated_mean）。

2.9 其他可视化方法

直方图和 PMF 在探索性数据分析中很有用：如果你对数据的含义有基本认识，设计一个能展示直观效应的可视化方法通常会有所帮助。

在 NSFG 数据中，明显的分布差异出现在众数附近。所以有必要放大图的这一部分，变换一下数据，从而突出差异。

图 2-3 中是 PMF 在 35 ~ 45 周间的差异。将结果乘以 100，表示差异的百分比。

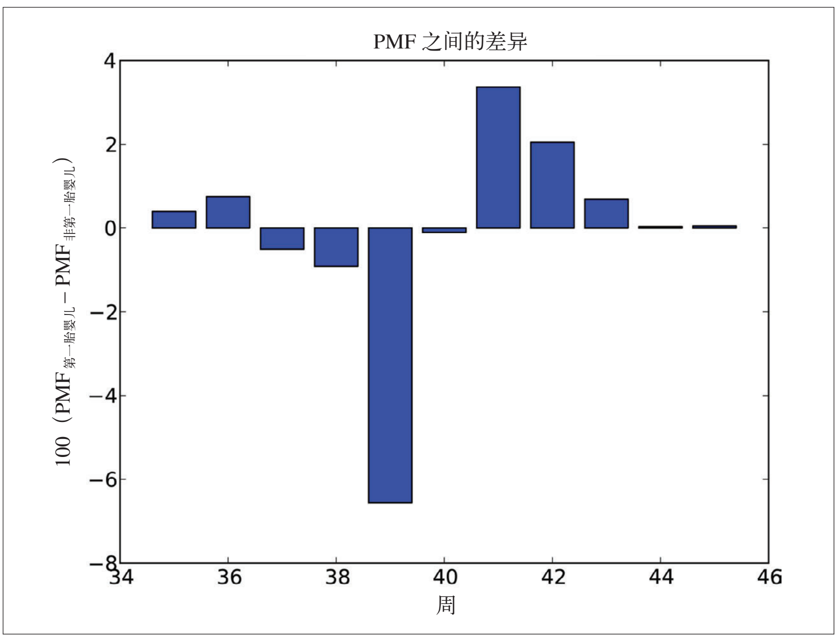


图 2-3：按周统计的婴儿出生百分比差异

这样分布趋势就非常清晰了：第一胎婴儿较少在 39 周出生，通常是在 41 周和 42 周出生。

2.10 相对风险

我们一开始就提出了这个问题：第一胎婴儿出生较晚吗？让我们明确一下，如果婴儿在第 37 周或更早出生，那就是提前出生；准时出生则是在第 38 到第 40 周；而延后出生则是在 41 周或更晚。这类用于数据分组的范围称为区间（bin）。

习题2-6

建一个 risk.py 文件。编写函数 `ProbEarly`、`ProbOnTime` 和 `ProbLate`，以 PMF 为参数，计算各个时间区间出生的婴儿所占的比例。提示：可以写一个通用函数实现这三个函数的功能。

准备三个 PMF，一个是第一胎婴儿的，一个是非第一胎婴儿的，还有一个是所有婴儿的。计算每个 PMF 中提前出生、准时出生和延后出生的婴儿的概率。

可以用相对风险（relative risk）来概括类似的数据，它代表两个概率的比值。例如，第一胎提前出生的概率是 18.2%。非第一胎婴儿提前出生的概率是 16.8%，因此相对风险就是 1.08。这意味着第一胎较其他几胎更早出生的可能性有 8%。

编写代码确认该结果，然后计算准时出生和延后出生的相对风险。可以从 http://thinkstats.com/risk.py 下载答案。

2.11 条件概率

假设你认识的某个人怀孕了，现在是第 39 周的开始，那么宝宝这一周出生的概率是多少？如果这是第一胎的话，答案会有什么变化？

我们可以通过计算条件概率来回答这些问题。所谓条件概率（conditional

probability）就是依赖于某个条件的概率。在这里，条件就是已知宝宝没有在前 38 周出生。

下面是一种计算方法。

1. 根据 PMF 生成一个 1000 名孕妇的模拟人群。对于每个周数 x，怀孕 x 周的孕妇人数为 1000PMF(x)。
2. 删除所有怀孕周数不足 39 的孕妇。
3. 计算余下怀孕周期的 PMF，这就是一个条件 PMF。
4. 计算 x=39 时条件 PMF 的值。

这个算法的概念很简单，但效率不高。一种简单的替代方案是将小于 39 的值从分布中删除，然后重新将数据归一化。

习题2-7

编写一个函数实现上述任意一个算法，计算宝宝在第 39 周出生的概率（假设宝宝没有在 39 周前出生）。

将该函数扩展成可以计算宝宝在任意第 x 周出生的概率（假设宝宝没有在 x 周之前出生）。画出第一胎宝宝和其他宝宝的该值与 x 的函数关系图。

答案可以从 http://thinkstats.com/conditional.py 下载。

2.12 汇报结果

至此，我们已经初步探索了数据，看到了一些直观效应。现在，假设这些结果都是真实的（但要记住，这只是一个假设），我们怎么汇报这些结果？

不同的人需要不同的答案。例如，科学家对实际效果更感兴趣，无论多小。医生可能只关注临床上有重要意义的结果，也就是会影响治疗决策的结果。而孕妇可能更关心与自身密切相关的东西，例如前一节

中介绍的条件概率。

如何汇报结果还取决于具体目的。如果是要证明某种影响的显著性，可以选择汇总统计量，如强调差异的相对风险。如果是要说服某个患者，则可以选择能反映特定情况下差异的统计量。

习题2-8

根据前面练习得到的结果，假设要对结果进行总结，判断第一胎婴儿是否出生较晚。

如果从晚间新闻获取一则故事，应该选择哪个汇总统计量？要让某位焦虑的患者放松心情，该选择哪些统计量？

最后，假设你是 *The Straight Dope*（http://straightdope.com）的作者 Cecil Adams，你的任务是回答“第一胎婴儿出生较晚吗？”这个问题。用本章中的结果写一段文字，清晰、准确地回答该问题。

2.13 术语表

- 区间（bin）
  将相近数值进行分组的范围。

- 集中趋势（central tendency）
  样本或总体的一种特征，直观来说就是最能代表平均水平的值。

- 临床上有重要意义（clinically significant）
  分组间差异等跟实践操作有关的结果。

- 条件概率（conditional probability）
  某些条件成立的情况下计算出的概率。

- 分布（distribution）
  对样本中的各个值及其频数或概率的总结。

- 频数（frequency）
  样本中某个值的出现次数。
- 直方图（histogram）
  从值到频数的映射，或者表示这种映射关系的图形。
- 众数（mode）
  样本中频数最高的值。
- 归一化（normalization）
  将频数除以样本数量得到概率的过程。
- 异常值（outlier）
  远离集中趋势的值。
- 概率（probability）
  频数除以样本数量即得到概率。
- 概率质量函数（Probability Mass Function，PMF）
  以函数的形式表示分布，该函数将值映射到概率。
- 相对风险（relative risk）
  两个概率的比值，通常用于衡量两个分布的差异。
- 分散（spread）
  样本或总体的特征，直观来说就是数据的变动有多大。
- 标准差（standard deviation）
  方差的平方根，也是分散的一种度量。
- 修剪（trim）
  删除数据集中的异常值。
- 方差（variance）
  用于量化分散程度的汇总统计量。

第3章

累积分布函数

3.1 选课人数之谜

在大部分美国高校，师生比例都在 1 : 10 左右。但学生经常会发现，一门课程的平均选修人数会超过 10。这其中有两个原因。

- 学生每学期通常会上4到5门课程,但一个教授通常只会教一两门课。
- 喜欢小班（选课人数少）的学生人数往往很少，而参加大班（选课人数多）的学生人数特别多。

第一点很明显（至少我指出后你就能明白），第二点就不是那么直观了。让我们看一个例子。某个学校某学期开设了 65 门课程，每门课程的选课人数分布如下：

人数	课程数
5~9	8
10~14	8
15~19	14
20~24	4
25~29	6
30~34	12
35~39	8
40~44	3
45~49	2

如果我们问院长，平均每门课程的选课人数是多少？他会构建一个PMF，计算出均值，然后告诉你平均每门课程有 24 个人选修。

但如果你找学生做调查，询问他们参加的课程有多少学生，然后计算平均值，所得到的每门课程的平均人数就会多不少。

习题3-1

按照院长的方法构建这些数据的 PMF，并计算均值。因为数据是分组的，所以可以用每组的中点值。

然后再从学生的角度来构建选课人数的分布，并计算均值。

假设想要得到学校每门课程选课人数的分布情况，但又无法从院长那里得到可信的数据。其中一种解决办法是随机选择一组学生，然后询问他们所选课程的上课人数。然后可以根据调查的结果计算出 PMF。

这个结果是有偏差的。因为选修人数多的课程会被过采样，所以在估计选课人数真实分布时要对观察到的分布做一个合适的变换。

编写一个 `UnbiasPmf` 函数，参数是观察值的 PMF，返回据此估计出的表示选课人数分布的 Pmf 对象。

答案可以从 http://thinkstats.com/class_size.py 下载。

习题3-2

在大部分的田径比赛中，选手都是同时出发的。如果跑得快，那么在比赛刚开始的时候会超过很多人，但在跑出几英里后你就会发现，周围都是跟你速度差不多的选手。

我第一次参加长跑（209 英里）接力时，注意到一个奇怪的现象：当我超过其他选手时，我会跑得更快；当其他选手超过我时，他们通常也会跑得更快。

一开始，我觉得速度的分布是两级分化的：速度快和速度慢的人都很多，但跟我速度差不多的人应该不多。

但随后我发现我的选择是有偏差的。这个比赛有两个特点：分阶段出发，不同的队伍出发时间也不同；此外，同一个队伍中选手的水平也参差不齐。

因此，选手在比赛道路上所处的位置与其速度和名次没有什么关系。在我开始跑时，我周围的参赛选手基本上是随机的。

那这其中的偏差来自何处？在整个比赛过程中，超过其他选手或者是被其他选手超过的概率跟选手间速度差异的大小是有关的。为什么？想想最极端的情况。如果我跟另外一个比赛选手的速度完全一样，那我们就不可能超过对方，也不可能被对方超过。如果某个选手跑得特别快，在我跑的过程中跑完了全程，那这位选手肯定会在某个地方超过我。

写一个 `BiasPmf` 函数，其参数是表示选手速度实际分布的 Pmf 和观察者的速度，返回值是一个新的 Pmf，表示其他选手相对观察者的速度分布。

用一般的道路比赛（不是接力赛）的数据测试函数。我写了一个程序读取马萨诸塞州 Dedham 的 James Joyce Ramble 一万米比赛的数据，并将每个选手的速度单位转换成 m/h。可以从 http://thinkstats.com/relay.py 下载这个程序。运行该程序，看看速度的 PMF。

现在假设你以 7.5 m/h 的速度参加这个比赛，计算你所观察到的选手的速度分布。可以从这里下载答案：http://thinkstats.com/relay_soln.py。

3.2 PMF的不足

如果要处理的数据比较少，PMF 很合适。但随着数据的增加，每个值的概率就会降低，而随机噪声的影响就会增大。

例如，假设我们对新生儿的体重分布感兴趣。在 NSFG 的数据中，变量 `totalwgt_oz` 以盎司[1]为单位记录了新生儿的体重。图 3-1 分别是第一胎婴儿和其他婴儿的体重 PMF。这也说明了 PMF 的一个不足之

注 1：英制计量单位，1 盎司 =28.350 克。——编者注

处：很难做比较。

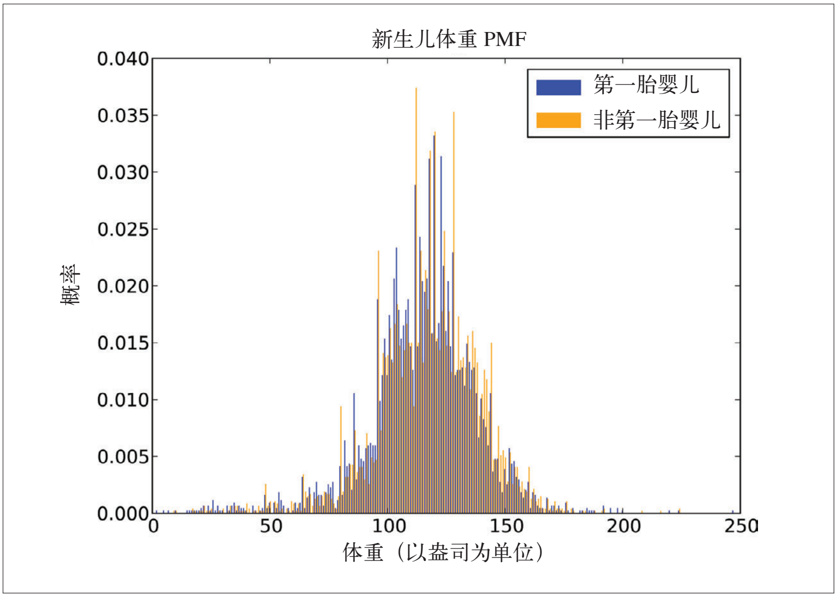


图 3-1：新生儿体重 PMF

整体上来看，这两个分布都是钟型曲线，均值附近的值比较多，远离均值的较大值和较小值都比较少。

但这个图中有些东西很难解读。其中有很多峰值和低谷，而且两个分布间有些很明显的差异。很难说哪些特征是显著的。此外，也不容易分辨整体的模式，比如哪个分布的均值比较大？

通过将数据分组可以解决这些问题。也就是将整个区域分成若干个不重叠的区间，然后计算每个区间内值的数量。分组很有用，但确定分组区间的大小就需要技巧了。分组区间大到能够消除噪声的时候，也会把有用的信息抹掉。

另一个解决这些问题的方法是*累积分布函数*（Cumulative Distribution Function，CDF）。不过在介绍 CDF 之前，我们先来说说*百分位数*（percentile）。

3.3 百分位数

标准化考试的成绩一般会以两种形式呈现，一种是原始分数，另一种则是*百分等级*（percentile rank）。在这里，百分等级就是原始分数不高于你的人在全部考试人数中所占的比例再乘以 100。所以，如果你在 90 百分位数，那就是说你比 90% 的人成绩好，或者至少不比 90% 的考试人员差。

下面的代码可以计算出给定值的百分等级，这里给定值是 `your_score`，所有分数是 `scores`：

```
def PercentileRank(scores, your_score):
    count = 0
    for score in scores:
        if score <= your_score:
            count += 1

    percentile_rank = 100.0 * count / len(scores)
    return percentile_rank
```

例如，如果这串分数是 55、66、77、88 和 99，而你的分数是 88。那么你的百分等级就是 $100 \times 4/5$，等于 80。

对于给定的值，很容易计算出百分等级，但反过来就要困难些。对于给定的百分等级，要找到对应的值，一种解决方法就是对所有的值排序，然后搜索想要的值：

```
def Percentile(scores, percentile_rank):
    scores.sort()
    for score in scores:
        if PercentileRank(scores, score) >= percentile_rank:
            return score
```

结果就是一个*百分位数*。例如，50 百分位数就是百分等级为 50 的值。在前面的考试分数分布中，50 百分位数就是 77。

习题3-3

前面实现的 `Percentile` 的效率并不高。一种更好的方法是用百

分等级计算相应的百分位数的索引。用这个算法实现一个新版的 `Percentile`。

答案可以从 http://thinkstats.com/score_example.py 下载。

习题3-4

选做：如果要计算某个百分位数，采用分数排序很低效。更好的解决办法是用选择算法，详见 http://wikipedia.org/wiki/Selection_algorithm。

实现选择算法，或者找一个现成的实现，用它编写一个更高效的 `Percentile` 函数。

3.4 累积分布函数

理解了百分位数，现在我们可以开始学习累积分布函数（CDF）了。CDF 函数就是值到其在分布中百分等级的映射。

CDF 是 x 的函数，其中 x 是分布中的某个值。计算给定 x 的 CDF(x)，就是计算样本中小于等于 x 的值的比例。

以下函数的参数是样本 `t` 和值 `x`：

```
def Cdf(t, x):
    count = 0.0
    for value in t:
        if value <= x:
            count += 1.0

    prob = count / len(t)
    return prob
```

读者应该很熟悉这个函数，这跟前面的 `PercentileRank` 函数几乎一样，唯一区别就是该函数返回的结果是 0 ～ 1 范围内的概率，而不是 0 ～ 100 范围内的百分等级。

来看个例子，给定一个样本 {1, 2, 2, 3, 5}。下面是其中某些值的 CDF：

CDF(0) = 0

CDF(1) = 0.2

CDF(2) = 0.6

CDF(3) = 0.8

CDF(4) = 0.8

CDF(5) = 1

我们可以计算任意值 x 的 CDF，而不仅是样本中出现的值。如果 x 比样本中最小的值还要小，那么 CDF(x) 就等于 0。如果 x 比样本中的最大值还要大，那么 CDF(x) 就是 1。

图 3-2 是这个 CDF 的图形化表示。样本的 CDF 是一个阶跃函数。在下一章，我们会看到作为连续函数的 CDF 的分布。

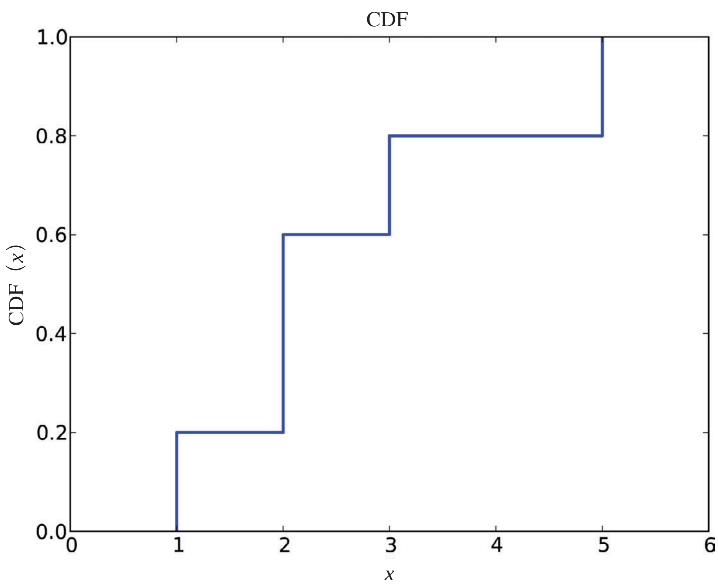


图 3-2：CDF 示例

3.5 CDF的表示

我编写了一个 `Cdf` 模块，提供了用于表示 CDF 的 `Cdf` 类。该模块的文档在 http://thinkstats.com/Cdf.html，也可以从 http://thinkstats.com/Cdf.py 下载。

Cdf 是用两个有序列表 `xs` 和 `ps` 实现的：其中 `xs` 列出了值，`ps` 列出了概率。Cdf 中最重要的方法如下所示。

- `Prob(x)`
  对于给定值 *x*，计算概率 *p*=CDF(*x*)。

- `Value(p)`
  对于给定概率 *p*，计算相应的值 *x*，也就是 *p*=CDF(*x*) 的逆运算。

因为 `xs` 和 `ps` 都经过了排序，所以上述操作都可以使用二分算法，效率很高。运行时间跟值数量的对数成正比，详见 http://wikipedia.org/wiki/Time_complexity。

Cdf 还有 `Render` 方法，它会返回 `xs` 和 `ps` 列表，可以用于绘制 CDF 图。因为 CDF 是一个阶跃函数，所以分布中的每个值都会在这两个列表中分别有对应的元素。

`Cdf` 模块还提供了几个生成 Cdf 的函数，包括 `MakeCdfFromList`，以一个序列为参数，返回它们的 Cdf。

最后要说一下，myplot.py 中的 `Cdf` 和 `Cdfs` 函数可以绘制 Cdf 折线图。

习题3-5

下载 Cdf.py 和 relay.py（参见习题 3-2），画出跑步速度的 CDF 图。哪个函数可以更好地看出分布形状，是 PMF 还是 CDF？代码可以从 http://thinkstats.com/relay_cdf.py 下载。

3.6 回到调查数据

图 3-3 是 NSFG 数据集中第一胎婴儿和其他婴儿出生体重的 CDF。

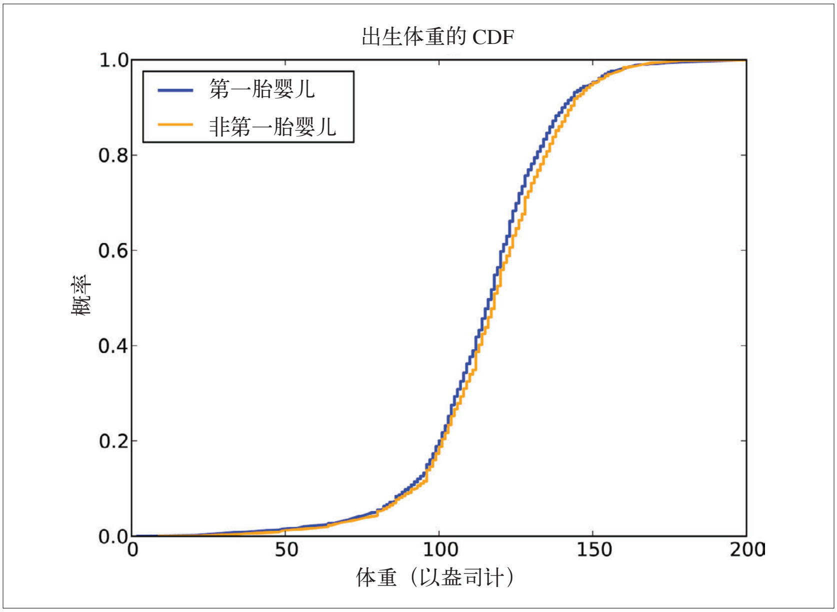


图 3-3：出生体重的 CDF

从这个图可以更加清楚地看出两个分布的形状和两者的差异。从两个分布可以看出，我们可以看到第一胎婴儿的体重略轻，在大于均值的部分有较大的差异。

习题3-6

你出生时有多重？如果你自己不知道的话，可以打电话给你妈妈或者是其他知道的人问问。利用所有活婴的数据计算婴儿出生体重数据分布，然后算算你在其中的百分等级。如果你是第一胎出生的，那再看看自己在第一胎婴儿的分布中的百分等级。如果不是第一胎出生的，那就看看你在其他婴儿中的百分等级。你在这两个分布中的百分等级有多大区别？

习题3-7

假如你和同学都计算了各自出生体重的百分等级，然后计算了百分等级的 CDF。你觉得这个分布会是什么样子？提示：预计班上有多少同学会在中位数以上。

3.7 条件分布

所谓条件分布就是根据某个条件选择的数据子集的分布。

例如，如果你的体重略高于平均值，但身高远远超过平均值，那么就你的身材来说，你可能体重偏轻。如何才能更精确地表述这个情况？

1. 选择一组身高跟你差不多（在一定范围内）的人。

2. 算出这群人体重的 CDF。

3. 找到你的体重在该分布中的百分等级。

比较来自不同测试的度量结果或不同分组的测试结果时，百分等级非常有用。

例如，参加田径比赛的人一般是按年龄和性别分组的。要比较不同分组中选手的水平，就可以将比赛时间转换成百分等级再做比较。

习题3-8

我最近参加了在马萨诸塞州 Dedham 举办的 James Joyce Ramble 一万米长跑。比赛结果放在了 http://coolrunning.com/results/10/ma/Apr25_27thAn_set1.shtml 上。访问这个网页可以看到我的比赛成绩：在全部 1633 名选手中排名第 97 位。那么我在所有参赛选手中的百分等级是多少？

在我所在的分组中（M4049 的意思是“40 到 49 岁之间的男性”），我在 256 名选手中排在第 26 位。我在这个分组中的百分等级是多少？

如果我在未来十年都参加这个比赛（希望我能行），我就会参加 M5059

分组。假设我在分组中的百分等级不变，我的速度会变慢多少？

我跟我的一个学生比成绩，她是 F2039 组的。她要在下次一万米长跑中跑多快才能在百分等级上赢我？

3.8 随机数

在生成服从给定分布的随机数时，CDF 是很有用的。

- 在 0 到 1 的范围内选择一个随机的概率。
- 用 `Cdf.Value` 找到你所选的概率在分布中对应的值。

这其中的原理似乎不太直观，但这容易实现，让我们动手试试。

习题3-9

编写一个 `Sample` 函数，参数是一个 Cdf 和一个整数 *n*，返回 *n* 个来自该 Cdf 的随机数。提示：用 `random.random`。在 Cdf.py 中有本练习的解答。

用来自 NSFG 的出生体重分布生成一个 1000 个元素的随机样本。计算该样本的 CDF。画出原始的 CDF 和随机样本的 CDF。如果 *n* 足够大，两个分布应该是一样的。

这个根据已有的样本生成随机样本的过程就称为*再抽样*（resampling）。

从总体获得样本的方法有两种：有放回和无放回。假设是从小桶子里面取玻璃球[2]，“有放回”就是在下次取球之前将之前取的球放回桶中（并搅动），所以每次取球后的总体都是不变的。“无放回”就是每个球只能取一次，这样每次取球的总体都不一样。

在 Python 中，可以用 `random.random` 选择百分等级或者用 `random.choice` 从序列中选择元素来实现有放回抽样，而用 `random.sample` 实现无放回抽样。

注 2：取球问题是随机抽样过程的标准模型（参见 http://wikipedia.org/wiki/Urn_problem）。

习题3-10

random.random 生成的数字介于 0 到 1 之间，也就是说该范围内的每个值被选中的概率是一样的。

用 random.random 生成 1000 个数字，画出它们的 PMF 和 CDF。是否能判断它们是均匀分布？

关于均匀分布，详见 http://wikipedia.org/wiki/Uniform_distribution_(discrete)。

3.9 汇总统计量小结

在算出 CDF 之后，再计算其他汇总统计量就比较容易了。中位数（median）就是百分等级是 50 的值。[3] 25 和 75 百分等级通常用来检查分布是否对称。这两者间的差异称为*四分差*（interquartile range），表示分布的分散情况。

习题3-11

编写一个 Median 函数，以 Cdf 作为参数，计算其中位数。再编写一个 Interquartile 函数，计算四分差。

计算出生体重 CDF 的 25、50 和 75 百分等级。从这些值是否可以判断分布是对称的？

3.10 术语表

- *条件分布*（conditional distribution）
  在满足一定前提条件下计算出的分布。

注 3：中位数还有其他定义。有些资料会说，当样本中元素的个数是偶数时，中位数是中间两个元素的平均值。没必要定义这种特殊情况，而且用一个样本中不存在的值会显得奇怪。本书中位数就是指百分等级为 50 的值。

- 累积分布函数（Cumulative Distribution Function，CDF）
  将值映射到其百分等级的函数。

- 四分差（interquartile range）
  表示总体分散情况的值，等于 75 和 25 百分等级之间的差。

- 百分位数（percentile）
  与百分等级相关联的数值。

- 百分等级（percentile rank）
  分布中小于或等于给定值的值在全部值中所占的百分比。

- 放回（replacement）
  在抽样过程中，“有放回”表示对于每次抽样，总体都是不变的。“无放回”表示每个元素只能选择一次。

- 再抽样（resampling）
  根据由样本计算得到的分布重新生成新的随机样本的过程。

第 4 章

连续分布

本书迄今为止所介绍的分布都是**经验分布**（empirical distribution），因为这些分布都是基于经验观察的，其中的样本都是有限的。

另一种分布是**连续分布**（continuous distribution），它的特点是其 CDF 是一个连续函数（跟阶跃函数完全不同）。很多实际现象都近似于连续分布。

4.1 指数分布

我首先介绍最简单的**指数分布**（exponential distribution）。举个例子，观察一系列事件之间的**间隔时间**（interarrival time）。若事件在每个时间点发生的概率相同，那么间隔时间的分布就近似于指数分布。

指数分布的 CDF 是：

$$\mathrm{CDF}(x)=1-\mathrm{e}^{-\lambda x}$$

参数 λ 决定了分布的形状。图 4-1 中是 $\lambda=2$ 时的 CDF。

通常，指数分布的均值是 $1/\lambda$，所以这个分布的均值是 0.5。分布的中位数是 $\log(2)/\lambda$，大概等于 0.35。

来看一个近似指数分布的例子，我们看看婴儿出生时间的间隔。1997 年 12 月 18 日，澳大利亚布里斯班的医院总共出生了 44 个婴儿[1]。这 44 个婴儿的出生时间数据在当地的历史文件中有记录，可以从 http://thinkstats.com/babyboom.dat 下载到这个数据。

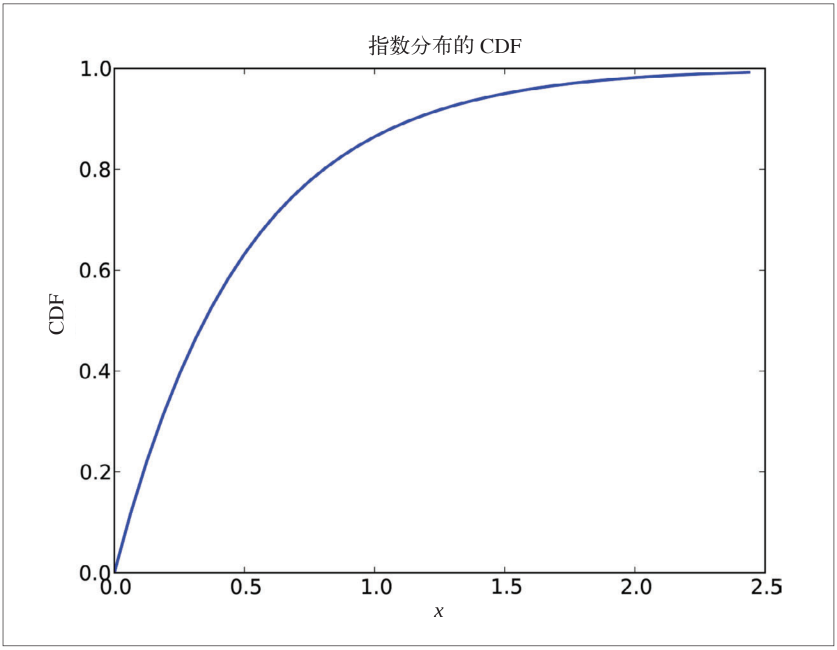


图 4-1：指数分布的 CDF

图 4-2 中是间隔时间的 CDF，单位是分钟。这看上去跟指数分布的形状很像，但我们如何才能确定这就是一个指数分布？

一种办法是画出取对数后的*互补累积分布函数*（Complementary CDF，CCDF）：1− CDF(x)。如果数据服从指数分布，这应该是一条直线。让我们看看为什么会这样。

注 1：这个例子的信息和数据来自 Dunn, "A Simple Dataset for Demonstrating Common Distributions," Journal of Statistics Education v.7, n.3 (1999)。

指数分布的数据集的 CCDF 如下：

$$y \approx e^{-\lambda x}$$

两边取对数得到：

$$\log y \approx -\lambda x$$

所以，在对 y 轴上的值取对数后，CCDF 是一条斜率为 $-\lambda$ 的直线。

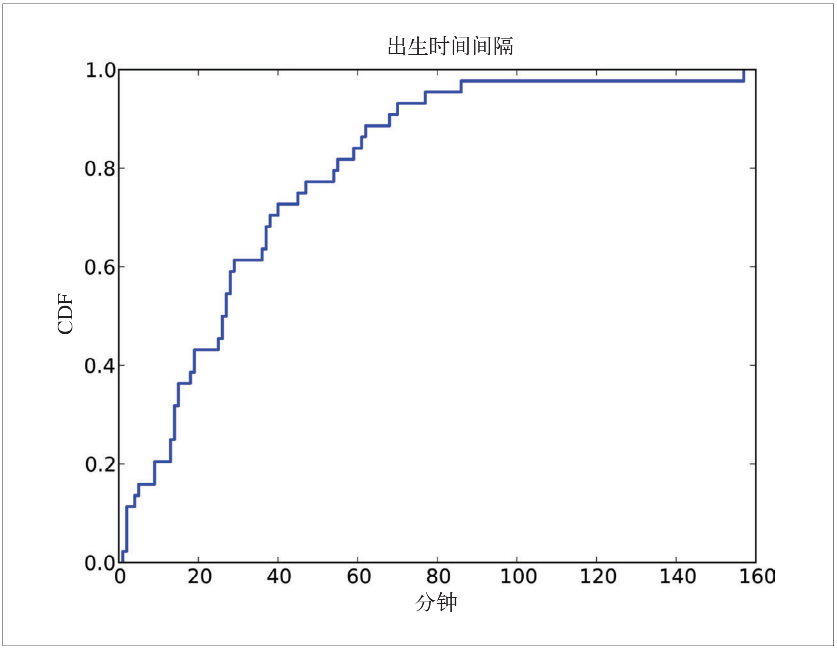


图 4-2：间隔时间的 CDF

图 4-3 是 y 轴取对数后的间隔时间的 CCDF。图中并不是一条严格意义上的直线，说明指数分布还只是一个近似。也就是说，以下假设并不完全正确：婴儿在一天中各个时间出生的概率一样。

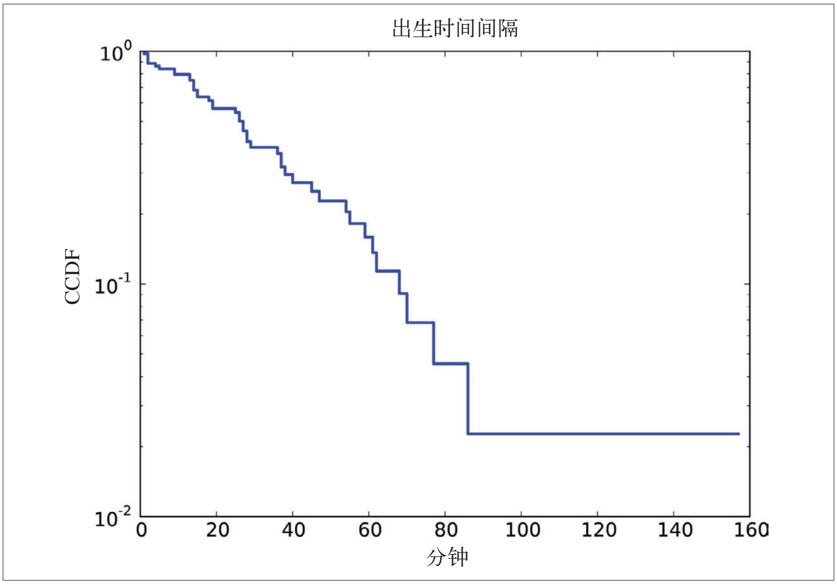


图 4-3：间隔时间 CCDF

习题4-1

在 *n* 较小时，经验分布不会很好地符合连续分布。评价两者间相似性的一个方法是从连续分布中生成样本，看看生成的样本跟数据的匹配情况。

`random` 模块中的 `expovariate` 函数可以为给定 λ 生成服从指数分布的随机数。用这个函数生成 44 个服从随机分布且均值为 32.6 的数。画出其 *y* 取对数后的 CCDF 图，将其与图 4-3 做比较。

提示：可以用 `pyplot.yscale` 画出取对数后的 *y* 轴。

也可以用 `myplot` 中的 `Cdf` 函数绘制 *y* 取对数后的 CCDF，`Cdf` 函数有一个选项 `complement`，用于判断是绘制 CDF 还是 CCDF，还有两个用于数轴转换的字符串选项 `xscale` 和 `yscale`：

```
myplot.Cdf(cdf,complement=True,xscale='linear',yscale='log')
```

习题4-2

收集你班上同学的生日，先排序，然后以天为单位计算同学生日的时间间隔。画出间隔时间的 CDF 和 y 轴取对数后的 CCDF，它们看上去像是指数分布吗？

4.2 帕累托分布

帕累托分布是以经济学家维尔弗雷多·帕累托（Vilfredo Pareto）的名字命名的，他曾用这个分布来描述财富分布情况（详见 http://wikipedia.org/wiki/Pareto_distribution）。从那以后，该分布就广泛用于描述自然界和社会科学中的各种现象，包括城镇大小、砂粒和陨石、森林火灾和地震等。

帕累托分布的 CDF 是：

$$\mathrm{CDF}(x) = 1 - \left(\frac{x}{x_m}\right)^{-\alpha}$$

参数 x_m 和 α 决定了分布的位置和形状。x_m 是最小值。图 4-4 是 x_m=0.5、α=1 的帕累托分布的 CDF。

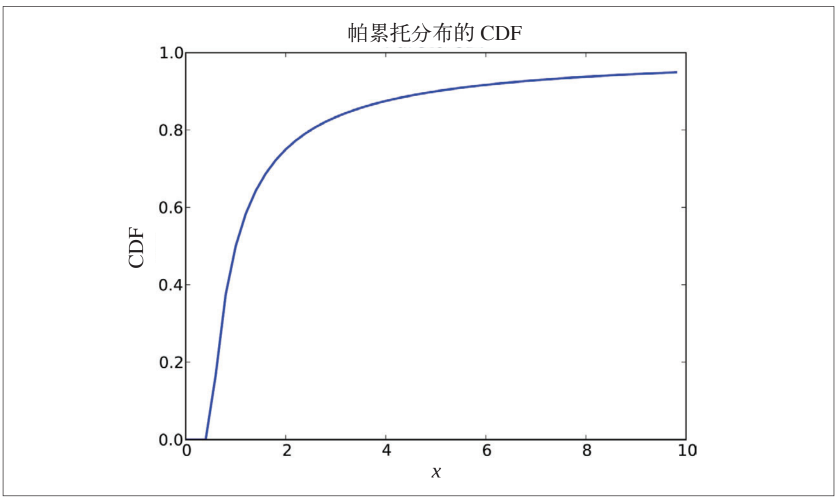


图 4-4：帕累托分布的 CDF

该分布的中位数是：

$$x_m 2^{1/\alpha}$$

即 1，但其百分等级为 95 的值是 10，而中位数为 1 的指数分布中百分等级为 95 的值仅仅是 1.5。

可以通过图形判断一个经验分布是否服从帕累托分布：对两条数轴都取对数后，其 CCDF 应该基本上是一条直线。如果直接画出服从帕累托分布的样本的 CCDF，其函数如下：

$$y \approx \left(\frac{x}{x_m}\right)^{-\alpha}$$

对两边取对数：

$$\log y \approx -\alpha(\log x - \log x_m)$$

在对 y 和 x 取对数后，就应该基本上是条直线，斜率是 $-\alpha$，截距是 $-\alpha \log x_m$。

习题4-3

`random` 模块中的 `paretovariate` 函数可以生成服从帕累托分布的随机数。该函数只有一个参数 α，却没有 x_m。x_m 默认值是 1，乘上 x_m，就可以生成各种不同的分布。

写一个 `paretovariate` 函数，以 α 和 x_m 作为参数，用 `random.paretovariate` 生成服从双参数帕累托分布的随机数。

用你自己编写的函数生成一个服从帕累托分布的样本。计算 CCDF，取对数后画出来。是一条直线吗？斜率是多少？

习题4-4

为了对帕累托分布有一个直观感受，让我们想象一下，如果全世界所有的人的身高服从帕累托分布会是一个什么情况。假设参数 x_m=100 厘米，α=1.7。这个分布的最小值是 100 厘米，而中位数是 150 厘米（这样比较合理）。

生成 60 亿个服从该分布的随机值。样本的均值是多少？其中有多大比例的人身高低于均值？在帕累托世界中最高的人多高？

习题4-5

Zipf 法则是一个关于各种单词使用频率差异的观察结论。常用单词的使用频率非常高，而罕见单词比如 hapaxlegomenon（一次频词）则使用得很少。 Zipf 法则说的是在一段文本（即语料库 corpus）中，单词频数的分布近似于帕累托分布。

找一个大的电子版语料库，任何语言的都可以。计算其中每个单词的出现次数。算出单词出现次数的 CCDF，画出取对数后的图。Zipf 法则是否成立？ α 的近似值是多少？

习题4-6

威布尔分布是一个广义上的指数分布，源自故障分析（failure analysis，详见 http://wikipedia.org/wiki/Weibull_distribution）。它的 CDF 是：

$$\mathrm{CDF}(x) = 1 - \mathrm{e}^{-(x/\lambda)^k}$$

能否找到某个变换将威布尔分布变成一条直线？这条直线的斜率和截距分别表示什么意思？

用 `random.weibullvariate` 生成一个服从威布尔分布的样本，然后用这个样本测试一下你的变换。

4.3 正态分布

*正态分布*也称为高斯分布，因其可以近似描述很多现象而成为最常用的分布。它的“普适性”是有原因的，我们会 6.6 节做介绍。

正态分布有很多使其适用于各种分析的特性，但 CDF 并不是其中之一。跟之前看到的分布不一样，我们对于正态分布的 CDF 还没有一种准确的表达。最常用的一种形式是以*误差函数*（error function）表示

的，误差函数是一种特殊函数，表示为 erf(x)：

$$\text{CDF}(x) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right]$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \text{e}^{-t^2}\text{d}t$$

参数 μ 和 σ 决定了分布的均值和标准差。

如果这些公式看起来眼花也别担心，在 Python 中很容易实现这些公式 [2]。有很多高效准确的方法来近似 erf(x)。我实现了一种，可从 http://thinkstats.com/erf.py 下载，其中提供了 `erf` 和 `NormalCdf` 两个函数。

图 4-5 是参数 μ=2.0，σ=0.5 的正态分布的 CDF。这种 S 型曲线就是正态分布的标志。

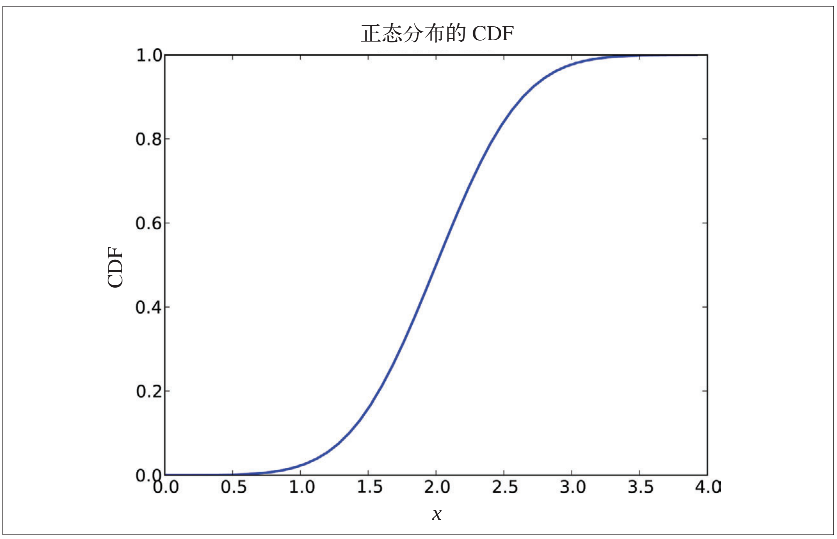


图 4-5：正态分布的 CDF

注 2：在 Python 3.2 中更容易，`math` 模块中就有 `erf` 函数。

在前一章中，我们看过 NSFG 数据中新生儿体重的分布。图 4-6 中是所有新生儿体重的经验 CDF，以及有着相同均值和方差的正态分布的 CDF。

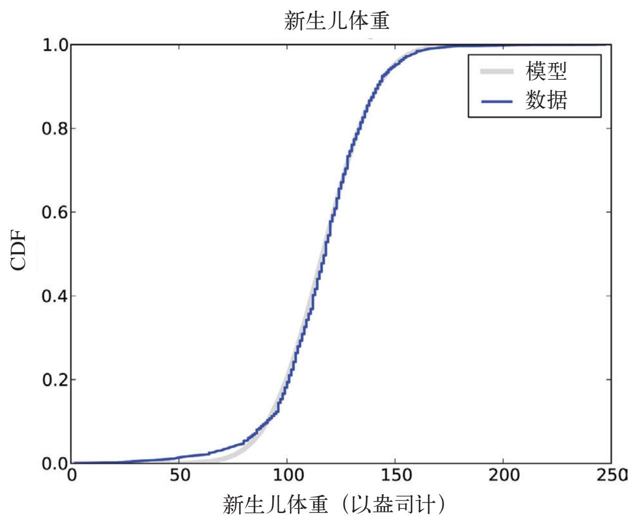


图 4-6：服从正态模型的新生儿体重的 CDF

对于这个数据集，正态分布是一个不错的模型。模型就是一种有效的简化。此处说它简单实用是因为我们可以用两个参数（μ=116.5 和 σ =19.9）来总结整个分布，并且所得到的误差（模型和数据之间的区别）很小。

在 10 百分位数以下的部分，数据和模型之间出现了差异，体重较轻的新生儿数量比我们根据正态分布得到的预期值要高一些。如果研究的是早产儿，那么这部分分布的正确性就非常重要，使用正态分布就不合适了。

习题4-7

The Wechsler Adult Intelligence Scale 是一个智商测试 [3]。我们对结果作变换，这样分数在一般人群中的分布就是正态的，参数为 μ=100 和 σ =15。

用 `erf.NormalCdf` 函数查看正态分布中罕见事件的频数。人群中有多大比例的人智商高于均值？高于 115、130、145 的分别是多少？

“六西格玛”事件就是超出均值 6 个标准差的值，所以六西格玛智商是 190。在全世界 60 亿人中，智商超过 190 的人有多少 [4]？

习题4-8

画出所有新生儿怀孕周期的 CDF。看上去像正态分布吗？

计算样本的均值和方差，根据这两个参数画出正态分布。使用该正态分布对数据建模合适吗？如果用两个统计量来总结这个分布，应该选哪两个统计量？

4.4 正态概率图

对于指数分布、帕累托分布和威布尔分布，都可以通过简单的转换来判断一个连续分布是否能用于某份数据集的建模。

然而，对于正态分布就不存在这样的变换，但有一种称为*正态概率图*（normal probability plot）的方法。它是基于秩变换（rankit）的，所谓*秩变换*就是对 n 个服从正态分布的值排序，第 k 个值分布的均值就称为第 k 个秩变换。

注 3：可以在闲暇时间了解一下，看看该智商测试是否吸引你，你觉得测试结果可靠吗？
注 4：这方面的详细信息请阅读 http://wikipedia.org/wiki/Christopher_Langan。

习题4-9

编写一个 `Sample` 函数，生成 6 个服从 μ=0，σ=1 的正态分布的值，将其排序后返回。

编写一个 `Samples` 函数，调用 1000 次 `Sample`，返回一个包含 1000 个列表的列表。

用 `zip` 函数处理这个列表组成的列表，可以得到 6 个包含 1000 个值的列表。计算每个列表的均值，输出结果。你得到的结果大概是这样：

{–1.2672, –0.6418, –0.2016, 0.2016, 0.6418, 1.2672}

随着调用 `Sample` 函数次数的增加，结果会收敛到这些值。

直接计算 rankit 是比较麻烦的，但有一些计算方法可以求出近似解。其中一个快捷且容易实现的方法如下。

1. 从 μ=0、σ=1 的正态分布中生成一个跟你的数据集大小一样的样本。

2. 将数据集中的值排序。

3. 画出数据集中排序后的值跟第一步生成的随机值的散点图。

对于大数据集，这个方法效果很好。对于较小的数据集，可以通过生成 $m(n+1)-1$ 个服从正态分布的值来提升效果，其中 n 是数据集的大小，而 m 是一个放大因子。然后从第 m 个元素开始选择第 m 个元素。

只要能生成所需的随机样本，该方法同样也可以用于其他的分布。

图 4-7 是一个简单的出生体重正态概率图。这个图的曲度表示数据集跟正态分布的差异；毕竟，在很多情况下，正态分布都是一个很好（至少是足够好）的模型。

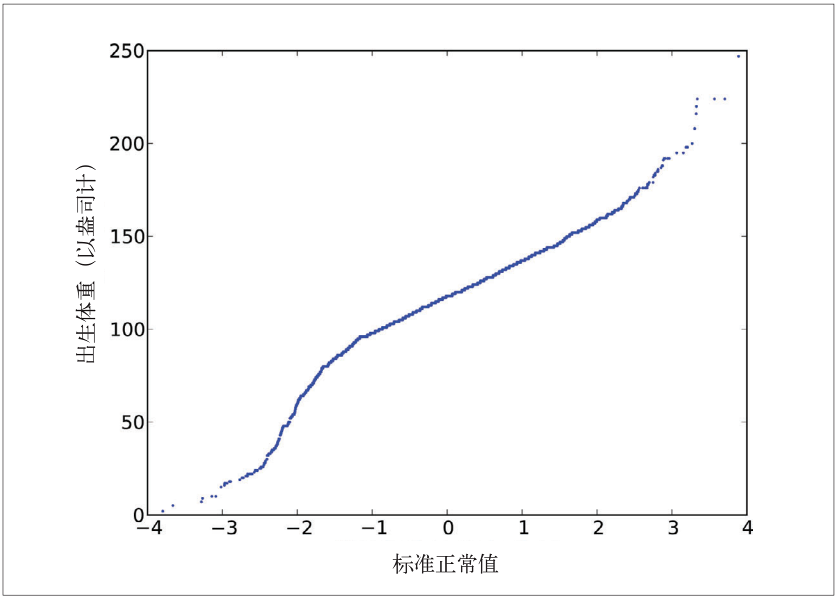


图 4-7：新生儿体重正态概率图

习题4-10

编写一个 `NormalPlot` 函数，输入是一个序列，生成一个正态概率图。答案在 http://thinkstats.com/rankit.py。

用 relay.py 中的跑步速度生成正态概率图。正态分布适用于这份数据吗？答案在 http://thinkstats.com/relay_normal.py。

4.5 对数正态分布

如果一组数值做对数变换后服从正态分布，我们就称其服从对数正态分布（lognormal distribution）。对数正态分布的 CDF 跟正态分布一样，只是用 log$x$ 代替原来的 x：

$$\mathrm{CDF}_{\mathrm{lognormal}}(x) = \mathrm{CDF}_{\mathrm{normal}}(\log x)$$

正态分布的参数通常用 μ 和 σ 表示。但要记住，这两个参数的意思不是均

值和标准差，对数正态分布的均值是 $\exp(\mu+\sigma^2/2)$，标准差则比较复杂[5]。

可以证明，成人体重的分布是近似对数正态的。[6]

美国国家慢性病预防和健康促进中心（NCCDPHP）每年都会进行一次调查，调查结果会作为行为风险因素监测系统（BRFSS）的一部分[7]。2008 年，他们访谈了 414 509 位被调查者，询问了他们的人口统计特征、健康和健康风险方面的问题。

这份调查数据中包含了 398 484 位被调查者的体重信息（单位是千克）。图 4-8 是 log*w* 的分布，其中 *w* 为体重，服从正态分布。

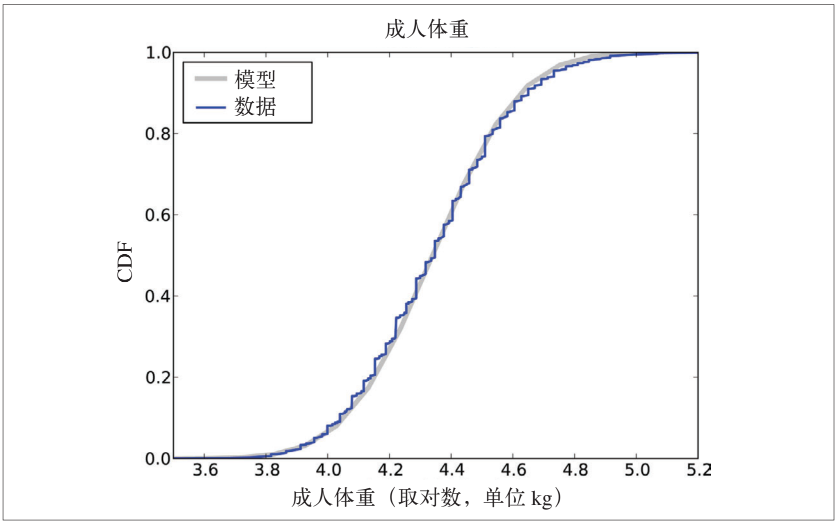


图 4-8：成人体重的 CDF（经过对数变换）

注 5：详见 http://wikipedia.org/wiki/Log-normal_distribution。

注 6：我是在 http://mathworld.wolfram.com/LogNormalDistribution.html 网页上看到这种观点的，但未注明来源。随后，我发现了一篇提出对数变换并解释其中原因的论文：Penman and Johnson, “The Changing Shape of the Body Mass Index Distribution Curve in the Population,” Preventing Chronic Disease, 2006 July; 3(3): A74。其网址是 http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1636707。

注 7：Centers for Disease Control and Prevention (CDC). Behavioral Risk Factor Surveillance System Survey Data. Atlanta, Georgia: U.S. Department of Health and Human Services, Centers for Disease Control and Prevention, 2008.

正态模型可以很好地拟合数据，但即使经过了对数变换，体重的最大值还是超过了我们的期望。而 *w* 取对数后符合正态分布，因此我们可以判断 *w* 服从对数正态分布。

练习4-11

从 http://thinkstats.com/CDBRFS08.ASC.gz 下载 BRFSS 数据，读取该数据的代码在 http://thinkstats.com/brfss.py 上。运行 brfss.py，应该可以看到它会在屏幕上显示出一些变量的汇总统计量。

写一个程序从 BRFSS 中读取成人体重，生成 *w* 和 log*w* 的正态概率图。答案可以从 http://thinkstats.com/brfss_figs.py 下载。

练习4-12

城镇人口分布是帕累托分布的一个真实例子。

美国人口普查局（U.S. Census Bureau）发布的人口数据涵盖全美所有城镇。我编写了一个小程序下载这些数据并将其保存到文件中，程序可以从 http://thinkstats.com/populations.py 下载。

1. 看一下程序，弄明白它能做什么，然后运行改该程序下载和处理数据。

2. 写个程序计算数据中 14 593 个城镇的人口分布，并画图。

3. 分别画出线性和取对数后的 CDF，看看分布的形状。然后两次取对数后画出 CCDF，看看该分布是否符合帕累托分布的特征。

4. 尝试本章中介绍的其他变换，画图，看看是否存在其他模型能更好地拟合这个数据。

关于城镇规模的分布，我们可以得出什么结论？相关代码可以从 http://populations_cdf.py 下载。

练习4-13

美国国税局（IRS）在 http://irs.gov/taxstats 上提供了关于个人所得税的数据。

其中一个文件中记录了 2008 年个人收入信息，网址是 http://thinkstats.com/08in11si.csv。我将其转换为逗号分隔文件（CSV），读者可用 csv 模块读取该文件。

从该数据集中抽取收入的分布。本章中介绍的连续分布是否能较好地拟合该数据？答案可从 http://thinkstats.com/irs.py 下载。

4.6 为什么需要模型

我在本章开头说过，真实世界的很多现象都可以用连续分布来建模。读者可能会问："这有什么用？"

跟所有模型一样，连续分布也是一种抽象。换言之，就是会舍弃一些无关紧要的细节。例如，真实观察到的分布中可能会有测量误差或是对样本来说很奇怪的数据，而连续模型会消除这些无关紧要的细节。

连续模型也是一种数据压缩。如果模型能很好地拟合数据集，那么少量参数就可以描述大量数据。

有时候，我们会惊讶地发现某种自然现象服从某个连续分布，观察这些现象可以让我们深入理解真实的系统。有时候，我们可以解释观察到的分布服从特定形式的原因。例如，帕累托分布通常是正反馈生成过程的结果（也称为偏好依附：preferential attachment，详见 http://wikipedia.org/wiki/Preferential_attachment）。

连续分布可用于数学分析，我们会在第 6 章中介绍。

4.7 生成随机数

连续分布 CDF 对于生成随机数也很有用。如果可以高效地计算出 ICDF(p)（inverse CDF，逆 CDF），我们就可以方便地生成服从各种分布的随机值。方法是首先产生 0~1 之间服从均匀分布的值，然后选择：

$$x = \mathrm{ICDF}(p)$$

例如，指数分布的 CDF 是：

$$p = 1 - e^{-\lambda x}$$

求解 x，得到：

$$x = -\log(1-p)/\lambda$$

用 Python 写成的代码如下所示：

```
def expovariate(lam):
    p = random.random()
    x = -math.log(1-p) / lam
    return x
```

我们用 `lam` 变量表示参数是因为 `lambda` 是 Python 关键字。大部分 `random.random` 实现都可以返回 0，但不能返回 1，所以 $1-p$ 有可能等于 1，但不可能等于 0，因为 log0 是没有定义的。

练习4-14

编写一个 `weibullvariate` 函数，参数是 `lam` 和 `k`，返回随机值，随机值服从以此为参数的威布尔分布。

4.8 术语

- 连续分布（continuous distribution）
  由连续函数描述的分布。

- 语料库（corpus）
  特定语言中用做样本的正文文本。

- 经验分布（empirical distribution）
  样本中值的分布。

- 误差函数（error function）
  一种特殊的数学函数，因源自误差度量研究而得名。

- 一次频词（hapaxlegomenon）
  表示语料库中只出现一次的词。这个单词在本书中迄今出现了两次。

- 间隔时间（interarrival time）
  两个事件的时间间隔。

- 模型（model）
  一种有效的简化。对于很多复杂的经验分布，连续分布是不错的模型。

- 正态概率图（normal probability plot）
  一种统计图形，用于表示样本中排序后的值与其服从正态分布时的期望值之间的关系。

- 秩变换（rankit）
  元素的期望值，该元素位于服从正态分布的已排序列表中。

第 5 章

概率

在第 2 章中，我们说过概率就是频数与样本数量的比值。这是概率的一种定义，但并不是唯一定义。实际上，概率的含义一直就是一个有争议的话题。

我们先搁置争议，看看其他内容。大家普遍认同概率是一个 0 到 1 之间的值，是一种定量度量，对应于定性地描述某一件事发生的可能性的大小。

被赋予概率的“事情”称为*事件*（event）。如果 E 表示一个事件，那么 $P(E)$ 就表示该事件发生的概率。检测 E 发生情况的过程就叫做*试验*（trial）。

举个例子，假设有一个标准的六面骰子，计算抛出 6 点的概率。每抛一次就是一次试验。抛出 6 点就是*成功*，否则就是*失败*。在某些情况下，“成功”可能是坏事，而“失败”才是好事。

如果我们有一个包含 n 次试验的有限样本，其中我们观察到 s 次成功，那么成功的概率就是 s/n。如果这个试验集合是无限的，概率的定义就会复杂些。但大部分人都可以接受用一系列假想的重复试验来表示概率，例如抛硬币或掷骰子。

在遇到不同事件的概率时，我们就遇到麻烦了。例如，我们想知道候选人赢得选举的概率。但每次选举都是不同的，因此不存在一系列的重复试验来计算概率。

遇到这种情况，有些人就会说上面这个概率的概念在这里并不适用。上面这种观点称为*频率论*（frequentism），就是用频率来定义概率。如果没有一系列相同的试验，那就不存在概率。

频率论在哲学上是没有错误的，但它却限制了概率的使用范围，只限于随机的物理系统（例如原子衰变）或因无法预测而被视做随机的系统（例如意外死亡）。任何涉及人为因素的情况都不适用。

还有一种观点是*贝叶斯认识论*（bayesianism），这种观点将概率定义为事件发生的可信度。根据这个定义，概率几乎能用于所有情况。贝叶斯概率的一个问题是它会受个体认知的影响：对于同一事件，不同的人会因为所掌握的信息不一样而对其发生的可信度有不同的判断。正因为如此，很多人认为贝叶斯概率要比频率概率更主观。

举个例子，他信·西那瓦成为泰国总理的概率有多大？频率学派会说这个事件没有概率，因为找不到一系列试验来验证这个问题。他信是否能成为总理跟概率没有关系。

反之，贝叶斯学派会根据其自己所掌握的信息赋予这个事件一个概率。例如，如果你知道 2006 年泰国发生了一次政变，并且你非常肯定时任总理的他信流亡国外，你就有可能将这个概率设为 0.1。这个值真正代表的是你的记忆出错的可能性，或者他信复职的可能性。

看一下维基百科，你就会知道他信不是泰国的总理（在我写这本书的时候）。有了这个信息，你可能就会将这个概率改成 0.01，反映了维基百科针对此事件出错的可能性。

5.1 概率法则

对于频率概率，我们可以推出不同事件概率关系的法则。其中最著名

的应该是：

$P(AB) = P(A)P(B)$ 警告：该法则并非在所有情况下都成立！

其中 $P(AB)$ 是事件 A 和事件 B 同时发生的概率。这个公式很好记，要记住的就是它的成立是有前提条件的，即事件 A 和事件 B 相互独立。换言之，就是事件 A 的发生对事件 B 发生的概率没有任何影响，反之亦然。

例如，如果事件 A 是抛硬币得到正面，而事件 B 是掷骰子得到一点，那么 A 和 B 两个事件就是相互独立的，因为抛硬币的结果跟掷骰子是没有任何关系的。

但如果我是掷两次骰子，事件 A 是至少得到一个六点，而事件 B 是得到两个六点，那 A 和 B 就不是独立的。因为，如果我知道事件 A 发生了，那事件 B 发生的概率就会上升；而如果我知道事件 B 发生了，那事件 A 发生的概率就是 1。

当事件 A 和 B 不独立时，通常需要计算条件概率 $P(A|B)$，即在事件 B 已经发生的情况下事件 A 发生的概率：

$$P(A\,|\,B) = \frac{P(AB)}{P(B)}$$

据此，我们可以得到更一般的关系：

$$P(AB) = P(A)P(B|A)$$

这个公式稍微难记点儿，但你把它用语言描述一下就非常好理解，“两件事情同时发生就是第一件事发生后第二件事情也发生了”。

事件发生的先后顺序是没有影响的，所以我们也可以这样写：

$$P(AB) = P(B)P(A|B)$$

无论事件 A 和 B 是否独立，这个关系都是成立的。如果它们是独立的，那么 $P(A|B) = P(A)$，我们就得到了一开始说的那个公式。

因为概率的范围是 0 到 1，所以很容易证明：

$$P(AB) \leqslant P(A)$$

想象一下，某俱乐部只接受符合特定要求 A 的人成为其会员。现在假设他们增加了一个新的要求 B，显然俱乐部的规模会变小，或者，如果所有成员都满足新要求，俱乐部规模保持不变。但有时候，人们极不擅长此类分析。关于这类现象的例子和相关讨论可以访问 http://wikipedia.org/wiki/Conjunction_fallacy。

习题5-1

掷两次骰子，总点数是八，那么其中一次是六点的概率是多少？

习题5-2

掷 100 次骰子，全部都是六点的概率是多少？没有六点的概率是多少？

习题5-3

下面的问题来自 Mlodinow 的 *The Drunkard's Walk* 一书。

1. 家里有两个小孩，都是女孩的概率是多少？
2. 家里有两个小孩，已知其中至少有一个女孩，两个都是女孩的概率是多少？
3. 家里有两个小孩，已知年龄较大的一个是女孩，两个都是女孩的概率是多少？
4. 家里有两个小孩，已知其中有一个叫 Florida 的女孩，两个都是女孩的概率是多少？

可以假设任意一个小孩是女孩的概率是 1/2（在很多方面都适用），而家里不同小孩的性别是独立事件。还可以假设女孩叫 Florida 的概率比较小。

5.2 蒙提霍尔问题

蒙提霍尔问题（The Monty Hall problem）有可能是历史上最富争议的概率问题。问题很简单，但正确答案与人们的直觉完全不同，所以很多人难以接受。不少聪明人不仅自己弄错了，还公开为错误的结果高声辩护。

蒙提·霍尔原本是美国电视游戏节目 *Let's Make a Deal* 的主持人，蒙提霍尔问题就是源自该节目中的一个游戏。如果你是参赛者，以下是节目现场的情况。

- 你会看到三扇关闭的门，蒙提会告诉你每扇门后的奖励：其中有一扇门后面是一辆车，而另外两扇门后面则是诸如花生酱或假指甲之类不太值钱的东西。奖品的摆放是随机的。
- 你的目标就是要猜出哪扇门后是汽车。如果猜对，汽车就归你了。
- 我们把你选择的门称为 A 门，其他两扇门分别是 B 门和 C 门。
- 在打开你所选择的 A 门之前，蒙提往往会打开 B 门或 C 门扰乱你的选择。（如果汽车确实是在 A 门后面，那蒙提随机打开 B 门或 C 门都没有问题。）
- 接下来，蒙提会给你一个机会：你是坚持原来的选择，还是选择另一扇未打开的门。

问题是，坚持原来的选择或选择另一扇门，会有什么不同吗？

大部分人凭直觉觉得这没有区别。因为，还剩下两扇门，所以汽车在 A 门后面的概率是 50%。

但这就错了。实际上，坚持选择 A 门，获胜的机会就只有 1/3；而如果选择另一扇门，获胜的机会就是 2/3。接下来我会解释原因，但信不信由你。

其中的关键在于要明白，这里有三种可能的情况：汽车可能会在 A 门后，也可能在 B 门或 C 门后面。因为奖品是随机摆放的，所以每种情

况的概率都是 1/3。

如果坚持选择 A 门，那么就只有在一开始汽车就在 A 门后面的情况下才能获胜，获胜的概率是 1/3。

但如果选择另一扇没打开的门，那么在 B 或 C 后面有车这两种情况下都会获胜，总体的获胜概率就是 2/3。

你可能还是不信，没关系，很多人都跟你一样。Paul Erdős 的一位朋友跟他解释这种情况时，他回答："不对，这绝不可能，两者没有区别。"[1]

穷尽各种解释都不能说服他。最终，只能用计算机模拟让他接受这个结果。

习题5-4

写一个模拟蒙提霍尔问题的程序，用这个程序估计坚持选择 A 门和选择另一扇门的获胜概率分别是多少。

然后阅读 http://wikipedia.org/wiki/Monty_Hall_problem 上关于此问题的讨论。

哪个更有说服力，是模拟还是各种解释？为什么？

习题5-5

理解蒙提霍尔问题的重点在于要明白蒙提打开一扇门实际上是给你提供了信息。想象一下，如果蒙提不知道奖品在哪里，随机打开 B 门或 C 门，会怎么样？

如果打开的门后是汽车，游戏结束，你输了，再没有选择的机会。否则，你是应该坚持还是改变选择？

注 1：参见 Hoffman 的 *The Man Who Loved Only Numbers* 一书的 83 页。

5.3 庞加莱

亨利·庞加莱（Henri Poincaré）是法国著名的数学家，1900 年左右在索邦大学任教。下面这个关于他的传闻可能是杜撰的，但这里有一个很有意思的概率问题。

庞加莱怀疑当地的面包屋出售的大面包的重量并没有他们所宣传的 1000 克那么重，所以他一年中每天都去买一个大面包，然后回家称重。到了年底，他画出了重量的分布图，并证明了该分布服从均值 950 克、标准差 50 克的正态分布。他把这个证据提交给了监管部门，监管部门警告了面包屋。

第二年，庞加莱继续每天测量他所购买的面包的重量。到了年底，他发现平均重量是 1000 克，跟宣传的一样。但他再次向监管部门投诉了面包屋，而这次监管部门处罚了面包屋。

为什么？因为重量分布是不对称的。跟正态分布不一样，这个分布向右倾斜，换言之就是面包屋做的面包依然只有 950 克，只是故意把比较重的面包卖给了庞加莱。

习题5-6

写一个程序模拟面包屋，从均值 950 克、标准差 50 克的分布中随机选出 n 块面包，把其中最重的一块给庞加莱。n 等于什么值会得到一个均值 1000 克的分布？标准差是多少？

将这个分布跟均值、标准差相同的正态分布作比较。两者形状上的差异是否足以说服监管部门？

习题5-7

如果跳舞时舞伴是随机安排的，女的比男的高的比例是多少？

在 BRFSS（见 4.5 节“对数正态分布”），身高的分布基本上是个正态分布，男性身高分布的 μ=178 cm，σ^2 =59.4 cm，女性身高分布的

μ=163 cm，σ^2 =52.8 cm。

插一句，读者可能发现男性身高的标准差比较大，这是否意味着男性身高的变化更大？比较两组的变化程度可以计算变异系数（coefficient of variation），即标准差除以均值，σ/μ。根据这个指标，女性身高的变化更大一些。

5.4 其他概率法则

如果两个事件是互斥的，即两者中只有一个会发生，那么两者的条件概率等于 0：

$$P(A|B)=P(B|A)=0$$

在这种情况下，计算任一事件的概率很容易：

$P(A \text{ 或 } B)=P(A)+P(B)$　警告：某些情况下是不成立的。

但要记住，只有在事件互斥的情况下这个公式才成立。在一般情况下，事件 A 发生，或 B 发生，或两者都发生的概率是：

$$P(A \text{ 或 } B)=P(A)+P(B)-P(AB)$$

减去 $P(AB)$ 是因为它被计算了两次。

例如，抛两枚硬币，至少一次反面朝上的概率是 1/2+1/2−1/4。如果不减去 1/4 的话，两次都正面朝上的概率就会被算两次。如果是抛三个硬币的话，这个问题会更清楚。

习题5-8

掷两次骰子，至少得到一个六点的概率是多少？

习题5-9

计算事件 A 或 B 发生，且两者不会同时发生的概率一般用什么公式？

5.5 二项分布

掷 100 次骰子，全部都是六点的概率是 $(1/6)^{100}$，而一个六点都没有的概率是 $(5/6)^{100}$。

这都很简单，但通常我们更想知道得到 k 个六点的概率，k 是 0 到 100 间的任意数。答案就是二项分布（binomial distribution），其 PMF 是：

$$\text{PMF}(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

其中 n 是试验总次数，p 是成功的概率，k 是成功的次数。

二项系数可以读作“n 中选 k”，可以直接计算出来：

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

也可以这样：

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

两个极端情况：如果 n=0，结果就是 0；如果 k=0，结果就是 1。访问 http://thinkstats.com/thinkstats.py，可以看到计算二项系数的 `Binom` 函数。

习题5-10

抛 100 次硬币，应该可以得到 50 次正面朝上，但恰好 50 次正面朝上的概率是多少？

5.6 连胜和手感

人们对随机过程的直觉往往跟事实有一定差距。如果要某人生成一些随机数，他可能会给你一些看上去随机，但实际上要比真正的随机数列有序得多的数字。反之，给他一个真正的随机数列，他也能从中找出一些并不存在的模式。

第二个现象的一个例子是很多人在体育运动中相信连胜或连败：大家往往认为一段时间比较成功的运动员“手感好”，而不成功的运动员则是“走霉运”。

统计学家在各种体育运动中测试了这些假设，但所有的结果都是一致的：不存在诸如连胜、连败一类的东西[2]。假设每次比赛都是独立事件，看到多次连胜或者连败的情况也很正常。这并不能说明这次获胜和下次获胜之间有什么联系。

另一个类似现象是聚类错觉（clustering illusion），指看上去好像有某种特点的聚类实际上是随机的（参见 http://wikipedia.org/wiki/Clustering_illusion）。

要检查某个聚类结果是否有意义，可以使用模拟随机系统，看看在随机情况下产生类似聚类的概率。这个过程就叫做蒙特卡罗模拟（Monte Carlo simulation），因为生成随机数的方法源自赌场（蒙特卡罗是有名的赌城）。

习题5-11

如果一场篮球比赛的 10 名参赛选手每人都投了 15 次篮，每次命中的概率是 50%，那么一场比赛中至少有一名球员投篮命中 10 次的概率是多少？另假设一个赛季是 82 场比赛，如果你看完整个赛季，那么至少看到一次连续 10 次命中或连续 10 次不命中的概率是多少？

这个问题说明了蒙特卡罗模拟的优缺点。其优点是编写模拟简单快速，不需要对概率有深入理解，缺点则是对于罕见事件的模拟需要很长的时间。稍做点儿分析可以省下大量的计算资源。

习题5-12

1941 年，Joe DiMaggio 在连续 56 场比赛中都有得分记录[3]。很多棒球

注 2：例如，参见 Gilovich、Vallone 和 Tversky 的“The hot hand in basketball: On the misperception of random sequences，”1985。

注 3：详见 http://wikipedia.org/wiki/Hitting_streak。

爱好者都觉得这是体育史上一项伟大的成就，因为这太少见了。

用蒙特卡罗模拟估计接下来的一个世纪中，棒球大联盟比赛中有球员在连续 57 场或更多场比赛中有得分记录的概率。

习题5-13

根据疾控中心（CDC）的定义，癌症聚集（cancer cluster）指的是“在一段时间内，某个地区的人群中的癌症病例高于预期值”。[4]

很多人觉得癌症聚集是环境恶化的证据，但很多科学家和统计学家觉得研究癌症聚集纯属浪费时间。[5] 为什么？其中一个原因就是癌症聚集是神枪手谬误的典型例子（Sharpshooter Fallacy，详见 http://wikipedia.org/wiki/Texas_sharpshooter_fallacy）。

不过，只要有人报告癌症聚集，CDC 还是有责任进行调查。根据他们的网页：

> 调查员先确定“病例”的定义，所关注的时间段，以及有风险的人群。然后计算预期值，并将其与实际观察到的值作比较。如果观察值和预期值的比值大于1且差异是统计显著的，就确认了存在聚集现象。

1. 假设某种癌症每年的发病率是千分之一。如果对 100 个人跟踪 10 年，应该能观察到一例病人。如果有两例也并不奇怪，但超过两例就比较少见了。
   写个程序模拟大量人群的 10 年期发病情况，估计出总病例数的分布。
2. 当某个观察值在完全随机的情况下出现的概率（即 p 值）小于 5% 时，我们就说它是统计显著的。在 100 个人历经 10 年的观察数据中，要出现多少病例才能满足这个要求？
3. 现在将 10 000 个人分为 100 个由 100 人组成的人群，跟踪 10 年。其中至少有一个人群出现“统计显著”聚集的概率是多少？如果把 p 值的要求改成 1% 呢？

注 4：源自 http://cdc.gov/nceh/clusters/about.htm。

注 5：参见 Gawande, “The Cancer Cluster Myth,” *New Yorker*, Feb 8, 1997。

4. 现在将 10 000 人放到 100 乘 100 的格子中，跟踪 10 年。其中至少有一个 10 乘 10 的方块出现统计显著聚集的概率是多少？

5. 最后，对方格中的 10 000 个人跟踪 30 年。其中某个 10 乘 10 的方块在某 10 年间隔中出现统计显著聚集的概率是多少？

5.7 贝叶斯定理

贝叶斯定理（Bayes’s theorem）描述的是两个事件的条件概率之间的关系。条件概率通常写成 $P(A|B)$，表示的是在事件 B 已发生的情况下事件 A 发生的概率。贝叶斯定理用公式表达如下：

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

为了说明这个公式的正确性，不妨把事件 A 和事件 B 同时发生的概率写作 $P(AB)$）：

$$P(AB)=P(A)P(B|A)$$

同样，

$$P(AB)=P(B)P(A|B)$$

也是成立的，因此：

$$P(B)P(A|B)=P(A)P(B|A)$$

两边都除以 $P(B)$ 就得到了贝叶斯定理[6]。

贝叶斯定理通常用于解释某一特定现象的证据 E 如何影响假设 H 的概率：

$$P(H|E) = P(H)\frac{P(E|H)}{P(E)}$$

这个等式的意思是：在看到 E 之后 H 的概率 $P(H|E)$，等于看到该证据前 H 的概率 $P(H)$，乘以假设 H 为真的情况下看到该证据的概率

注 6：详见 http://wikipedia.org/wiki/Q.E.D.。

$P(E|H)$ 与在任何情况下看到该证据的概率 $P(E)$ 的比值 $P(E|H)/P(E)$。

这种解读贝叶斯定理的方法叫做“历时性”（diachronic）解读，因为这描述了假设成立的概率随时间发生的变化，这种变化通常都是因为出现了新证据。在这种情况下，$P(H)$ 称为先验概率（prior probability），而 $P(H|E)$ 称为后验概率（posterior probability）。$P(E|H)$ 是证据的似然值（likelihood of evidence），$P(E)$ 是归一化常量（normalization constant）。

贝叶斯定理的一个经典应用就是解读临床检测。例如，学校和工作单位违禁药物检查正变得越来越普遍（详见 http://aclu.org/drugpolicy/testing）。采用这些检查的用人单位希望这些检查是敏感的，即当样品中有药物（或相关代谢物）时要得到一个阳性结果，而且要是特异的，即当样品中没有药物时得到阴性结果。

根据美国医学协会杂志（*Journal of the American Medical Association*）的估计[7]，常规药检的灵敏度大约是 60%，特异性大概是 99%。

现在假设这种常规检查适用于实际使用违禁药物比例为 5% 的职工人群，那么检查结果为阳性的雇员中有多少真正使用了违禁药物？

在贝叶斯理论中，我们要计算的就是当检查结果为阳性时，用药的概率 $P(D|E)$。根据贝叶斯定理：

$$P(D\,|\,E) = P(D)\frac{P(E\,|\,D)}{P(E)}$$

先验概率 $P(D)$ 就是我们看到检查结果之前用药的概率（即 5%），似然值 $P(E|D)$ 就是使用了违禁药的情况下检查结果为阳性的概率（即灵敏度）。

计算归一化常量 $P(E)$ 会稍微困难点。要考虑两个概率，$P(E|D)$ 和 $P(E|N)$，其中 N 代表受试者没有用药的假设：

注 7：我是从 Gleason 和 Barnum 的“Predictive Probabilities In Employee Drug-Testing”一文中看到这些数据的，网址是 http://piercelaw.edu/risk/vol2/winter/gleason.htm。

$$P(E)=P(D)P(E|D)+P(N)P(E|N)$$

假阳性的概率 $P(E|N)$ 跟特异性是互补的，即 1%。把这些放在一起，我们得到：

$$P(D\,|\,E)=\frac{P(D)P(E\,|\,D)}{P(D)P(E\,|\,D)+P(N)P(E\,|\,N)}$$

代入相应的值，我们得到 $P(D|E)=0.76$，这意味着每四个检查结果为阳性的人中大概有一个是被冤枉的。

习题5-14

写个程序，输入参数是用药的实际比例、监测的灵敏度和特异性，然后根据贝叶斯定理计算 $P(D|E)$。

假设将同样的检测用到用药比例为 1% 的人群，检查结果为阳性的人真正用药的概率是多少？

习题5-15

这个练习来自 http://wikipedia.org/wiki/Bayesian_inference。

> 假设有两碗饼干。一个碗中有10片巧克力饼干和30片普通饼干，而另一个碗中巧克力饼干和普通饼干各有20片。我们的朋友Fred随机选了一个碗，然后随机地从中取出一片饼干。那么Fred从第一个碗中取出普通饼干的概率是多少？

习题5-16

蓝色的 M&M 巧克力豆是 1995 年上市的。在那之前，一袋混合 M&M 巧克力豆的组成是 30% 棕色、20% 黄色、20% 红色、10% 绿色、10% 橙色和 10% 褐色；1995 年之后，其组成是 24% 蓝色、20% 绿色、16% 橙色、14% 黄色、13% 红色和 13% 棕色。

我的一个朋友有两袋 M&M，其中一袋是 1994 年产的，另一袋是 1996 年产的。但他没有告诉我到底哪一袋是 1994 年的，哪一袋是

1996 年的。现在朋友从两袋中各取出一粒巧克力豆，一粒是黄色，一粒是绿色。黄色巧克力豆来自 1994 年那袋的概率是多少？

习题5-17

这个练习改编自 MacKay 的 *Information Theory, Inference, and Learning Algorithms* 一书。

猫王 Elvis Presley 有一个双胞胎兄弟，但不幸在出生时就夭折了。根据维基百科上对双胞胎的介绍：

> 双胞胎占全世界人口的比例大概是1.9%，同卵双胞胎的比例大概是0.2%，占所有双胞胎的8%。

那么，Elvis 与夭折的兄弟为同卵双胞胎的概率是多少？

5.8 术语

- 贝叶斯认识论（Bayesianism）
  一种对概率更泛化的解释，用概率表示可信的程度。

- 变异系数（coefficient of variation）
  度量数据分散程度的统计量，按集中趋势归一化，用于比较不同均值的分布。

- 事件（event）
  按一定概率发生的事情。

- 失败（failure）
  事件没有发生的试验。

- 频率论（frequentism）
  对概率的一种严格解读，认为概率只能用于一系列完全相同的试验。

- 独立（independent）
  若两个事件之间相互没有影响，就称这两个事件是独立的。

- 证据的似然值（likelihood of the evidence）
  贝叶斯定理中的一个概念，表示假设成立的情况下看到该证据的概率。

- 蒙特卡罗模拟（Monte Carlo simulation）
  通过模拟随机过程计算概率的方法（详见 http://wikipedia.org/wiki/Monte_Carlo_method）。

- 归一化常量（normalizing constant）
  贝叶斯定理中的分母，用于将计算结果归一化为概率。

- 后验（posterior）
  贝叶斯更新后计算出的概率。

- 先验（prior）
  贝叶斯更新前计算出的概率。

- 成功（success）
  事件发生了的试验。

- 试验（trial）
  对一系列事件是否可能发生的尝试。

- 更新（update）
  用数据修改概率的过程。

第 6 章

分布的运算

6.1 偏度

偏度（skewness）是度量分布函数不对称程度的统计量。对于一个给定的序列 x_i，样本偏度的定义为：

$$g_1 = m_3 / m_2^{3/2}$$

$$m_2 = \frac{1}{n} \sum_i (x_i - \mu)^2$$

$$m_3 = \frac{1}{n} \sum_i (x_i - \mu)^3$$

这里 m_2 是均方离差（即方差），m_3 是平均的立方离差。

负的偏度表示分布向左偏（skews left），此时分布函数的左边会比右边延伸得更长；正的偏度表示分布函数向右偏。

上述计算样本偏度的公式在实际应用中使用得并不多。因为如果样本中存在异常值，那么这些异常值可能对偏度的值产生非常大的影响。

另外一个评价分布函数非对称程度的方法是比较均值和中位数的大

小。相比于中位数而言，均值更容易受极端值的影响，所以如果一个分布函数是向左偏的，那么该分布的均值就会小于中位数。

皮尔逊中值偏度系数（Pearson’s median skewness coefficient）就是一个基于这种思想的偏度度量（其中 μ 为均值，$\mu_{1/2}$ 为中位数）：

$$g_p = 3(\mu - \mu_{1/2})/\sigma$$

该统计量是偏度的一个鲁棒估计，它对异常值的影响不敏感。

习题6-1

请编写一个 `Skewness` 函数，计算一组样本数据的 g_1。

请计算怀孕周期和出生体重分布的偏度，这两个结果是否与分布的形状一致？

请编写一个 `PearsonSkewness` 函数，并用这个函数计算这些分布的 g_p。请比较 g_p 和 g_1 计算结果的差别。

习题6-2

乌比冈湖效应[1]（Lake Wobegon effect）是一种有趣的心理学现象，也称虚幻的优越性（ illusory superiority），是指人们通常会觉得自己各方面的能力都比社会上的平均水平高的一种心理倾向。例如，在一些研究中，超过 80% 的受调查者认为他们的驾驶技术高于平均水平（参见 http://wikipedia.org/wiki/Illusory_superiority ）。

假如社会平均水平指的是中位数，那么上述结果在逻辑上是不可能出现的。但是如果我们将平均水平定义为均值，那么上述结果就有可能出现，虽然可能性不大。

想想，长两条腿以上的人会占总人口的多少呢？

注 1：参见 http://wikipedia.org/wiki/Lake_Wobegon。

习题6-3

美国国税局（IRS）在其网站 http://irs.gov/taxstats 上提供了包括收入所得税在内的一些统计数据。如果做过习题 4-13，你应该已经接触过这些数据；如果没有做过，那么请按习题 4-13 的说明从数据集中提取收入的分布信息。

请问有多大比例的人申报的应纳税收入低于均值？

请计算收入数据的均值、中位数、偏度和皮尔逊中值偏度系数。由于数据已经按一定的区间进行了划分，这里的结果是一些近似值。

基尼系数（Gini coefficient）是一个衡量收入不平衡程度的指标。参考 http://wikipedia.org/wiki/Gini _coefficient 的信息编写一个名为 `Gini` 的函数，用于计算收入分布的基尼系数。

提示：可用 PMF 计算相对平均差（relative mean different），参考 http://en.wikipedia.org/wiki/Mean _difference 。

可以从这里下载到该问题的参考答案 http://thinkstats.com/gini.py 。

6.2 随机变量

随机变量（random variable）代表产生随机数的过程。随机变量一般用大写字母表示，如 X。当你看到一个随机变量时，可以把它想成从某个分布函数抽出来的值。

累积分布函数（cumulative distribution function）的形式化定义为：

$$\mathrm{CDF}_X(x) = P(X \leqslant x)$$

到目前为止，我一直在尽量避免使用这类数学符号，因为这些符号总有点难以理解。这个公式对累积分布函数进行了定义：随机变量 X 的累积分布函数在某个特定的值 x 上的函数值被定义为随机变量 X 小于等于 x 的概率。

从计算机人士的角度看，我们可以将随机变量设想成一个提供方法的

对象，将该方法命名为 generate，它能利用随机过程产生一些值。

例如，下面的代码可以用来表示随机变量的一个类：

```
class RandomVariable(object):
    """Parent class for all random variables."""
```

一个服从指数分布的随机变量：

```
class Exponential(RandomVariable):
    def __init__(self, lam):
        self.lam = lam

    def generate(self):
        return random.expovariate(self.lam)
```

init 方法接受一个参数 λ 并将其作为属性存储起来，generate 返回了一个服从参数为 λ 的指数分布的随机数。

每调用一次 generate 都将得到一个不同的数值。得到的数值称为随机数（random variate）。这就是在 random 模块中很多函数的名字都包含 variate 的原因。

如果仅仅是为了产生服从指数分布的随机数，或许不用费心地定义一个新类，可能直接用 random.expovariate 实现就行了。但是对于其他分布，用 RandomVariable 对象会是更好的选择。例如，爱尔朗分布（Erlang distribution）是一个连续型分布，有两个参数 λ 和 k。（参考 http://wikipedia.org/wiki/Erlang_distribution）。

一种产生服从爱尔朗分布的随机数的方法是，将 k 个服从参数为 λ 的指数分布的随机数进行求和。这里是它的一个实现：

```
class Erlang(RandomVariable):
    def __init__(self, lam, k):
        self.lam = lam
        self.k = k
        self.expo = Exponential(lam)

    def generate(self):
        total = 0
        for i in range(self.k):
```

```
            total += self.expo.generate()
        return total
```

init 方法产生一个给定参数的指数分布对象，generate 可以调用它。通常，init 方法可以接受任意一组参数，而 generate 函数则能实现任意随机过程。

习题6-4

请编写一个服从耿贝尔分布（Gumbel distribution）的随机变量的类的定义。(耿贝尔分布请参考 http://wikipedia.org/wiki/Gumbel_distribution。)

6.3 概率密度函数

累积分布函数的导数称为*概率密度函数*（probability density function），简记为 PDF。例如，指数分布的概率密度函数为：

$$\mathrm{PDF}_{\mathrm{expo}}(x) = \lambda \mathrm{e}^{-\lambda x}$$

正态分布的概率密度函数为：

$$\mathrm{PDF}_{\mathrm{normal}}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

概率密度函数的值并不是概率，它表示的是一种*概率密度*（probability density）。

物理学中，密度指的是单位体积内物质所拥有的质量，质量就等于体积乘以密度。如果物体的密度不固定，即不同部位密度不同，那么我们可以通过对密度函数进行积分得到质量。

同样地，概率密度衡量的是 x 轴上每个单位的概率。为了得到随机变量落在某个区间的概率[2]，我们可以计算其密度函数在这段区间上的积分。假设随机变量 X 的概率密度函数为 PDF_X，那么 X 落在区间 $[-0.5, 0.5)$ 的概率为：

$$P(-0.5 \leqslant X < 0.5) = \int_{-0.5}^{0.5} \mathrm{PDF}_X(x)\,\mathrm{d}x$$

注 2：我们可以解释得更具体一些，分布的均值是质心，方差是转动惯量。

因为累积分布函数是概率密度函数的积分，所以上式也可以写成：

$$P(-0.5 \leqslant X < 0.5) = \mathrm{CDF}_X(0.5) - \mathrm{CDF}_X(-0.5)$$

对有些分布，我们能够得到累积分布函数的解析表达式，这时可以采用第二种表达方式。对于一些无法得到累积分布函数解析表达式的分布，可以用数值积分的方法来计算 X 在某个区间上的概率。

习题6-5

假设一个随机变量 X 服从参数为 λ 的指数分布，那么 X 落在区间 [1, 20] 的概率是多少？请将结果表示成一个关于 λ 的函数，并保留推导结果，我们将在 8.8 节用到。

习题6-6

从 BRFSS（见 4.5 节）的数据中，我们发现人类身高大致服从正态分布，男性身高的均值为 178 cm、方差为 59.4 cm；女性身高的均值为 163 cm、方差为 52.8 cm。

蓝人乐团要求成员为男性，身高介于 178 cm 和 185 cm 之间，那么在美国男性中身高介于该区间的人有多少？

6.4 卷积

设两个随机变量 X 和 Y 的累积分布函数分别为 CDF_X 和 CDF_Y，$Z=X+Y$，那么 Z 服从什么分布呢？

一种简单的估算 Z 的分布的方法是编写一个 RandomVariable 对象，产生一些 X 和 Y，然后将它们加起来。下面是这种方法的代码实现：

```
class Sum(RandomVariable):
    def __init__(X, Y):
        self.X = X
        self.Y = Y

    def generate():
        return X.generate() + Y.generate()
```

我们可以产生很多 Z 的随机数，然后估计 CDF_Z。

上述方法是一个非常简单直观的方法，但不是很有效。为了能对 CDF_Z 有一个较准确的估计，我们必须产生大量的 X 和 Y 的随机数。而且不管这个精度被提高到什么程度，它总归不是 Z 的真实分布。

假设随机变量 X 和 Y 的累积分布函数 CDF_X 和 CDF_Y 有解析表达式，在某些情况下我们可以通过数学推导得到 CDF_Z 的解析表达式。

接下来介绍公式推导过程。

1. 从最简单的情况出发，假设随机变量 X 只能某个值 x，这时 $CDF_Z(z)$ 等于

   $$P(Z \leqslant z \mid X=x) = P(Y \leqslant z-x)$$

   上述等式的左半部分表示的是“在给定 $X=x$ 的条件下，$X+Y$ 小于 z 的概率”。显然，要使 $X+Y$ 小于 z，就要求 Y 必须小于 $z-x$。

2. 接下来，我们计算 Y 小于 $z-x$ 的概率，根据 Y 的累积分布函数有

   $$P(Y \leqslant z-x) = CDF_Y(z-x)$$

3. 上述推导过程假设 X 取某个固定值，但实际上它是一个随机变量。所以，我们还必须考虑 X 在其取值范围内的所有情况，于是：

   $$P(Z \leqslant z) = \int_{-\infty}^{\infty} P(Z \leqslant z \mid X = x) PDF_X(x) dx$$

   被积函数等于“给定 $X=x$ 的条件下，随机变量 Z 小于等于 z 的概率乘以 $X=x$ 的概率”。

   将前面两步的结果带入公式，有

   $$P(Z \leqslant z) = \int_{-\infty}^{\infty} CDF_Y(z-x) PDF_X(x) dx$$

   等式左边就是 CDF_Z 的定义，整理可得：

   $$CDF_Z(z) = \int_{-\infty}^{\infty} CDF_Y(z-x) PDF_X(x) dx$$

4. 我们可以通过对 CDF_Z 求导得到 PDF_Z，结果如下：

$$\mathrm{PDF}_Z(z) = \int_{-\infty}^{\infty} \mathrm{PDF}_Y(z-x)\mathrm{PDF}_X(x)\,\mathrm{d}x$$

如果读者之前学过信号与系统等课程，或许对这个积分公式不会感到陌生。它表示的是概率密度函数 PDF_X 和 PDF_Y 的卷积（convolution）。卷积运算一般用运算符 * 表示。

$$\mathrm{PDF}_Z = \mathrm{PDF}_Y * \mathrm{PDF}_X$$

综上可知，两个随机变量的和的分布就等于两个概率密度的卷积。参考 http://wiktionary.org/wiki/booyah!

下面我们通过一个例子来说明如何计算。设随机变量 X 和 Y 服从参数为 λ 的指数分布，则随机变量 $Z=X+Y$ 的概率密度为：

$$\mathrm{PDF}_Z(z) = \int_{-\infty}^{\infty} \mathrm{PDF}_X(x)\mathrm{PDF}_Y(z-x)\,\mathrm{d}x = \int_{-\infty}^{\infty} \lambda \mathrm{e}^{-\lambda x} \lambda \mathrm{e}^{\lambda(z-x)}$$

因为 X 和 Y 服从指数分布，它们取负数的概率为 0，所以我们可以对上述积分公式的下限进行调整：

$$\mathrm{PDF}_Z(z) = \int_0^z \lambda \mathrm{e}^{-\lambda x} \lambda \mathrm{e}^{-\lambda(z-x)}\mathrm{d}x$$

整理可得：

$$\mathrm{PDF}_Z(z) = \lambda^2 \mathrm{e}^{-\lambda z}\int_0^z \mathrm{d}x = \lambda^2 z \mathrm{e}^{-\lambda z}$$

这是参数为 k 和 λ 的爱尔朗分布的概率密度函数，这里 $k=2$。所以两个有相同参数的指数分布的卷积是一个爱尔朗分布。爱尔朗分布参考 http://en.wikipedia.org/wiki/Erlang_distribution。

习题6-7

假设随机变量 X 服从参数为 λ 的指数分布，Y 服从参数为 k 和 λ 的爱尔朗分布，$Z=X+Y$，那么 Z 服从什么分布呢？

习题6-8

假设我们从一个分布中抽取两个样本 X_1 和 X_2，$Y=\max(X_1, X_2)$，那么 Y 服从什么分布呢？请分别用 PDF 和 CDF 的形式表示。

随着值数量的增多，最大值的分布会收敛到某个极值的分布，参考 http://wikipedia.org/wiki/Gumbel_distribution。

习题6-9

假设我们有随机变量 X 和 Y 的 `Pmf`，$Z=X+Y$，那么就可以通过枚举法得到 Z 所有可能的取值：

```
for x in pmf_x.Values():
    for y in pmf_y.Values():
        z = x + y
```

请编写一个关于 PMF_X 和 PMF_Y 的函数，用于计算 Z 的 PMF。

请再编写一个类似的函数，用于计算 $\max(X,Y)$ 的 PMF。

6.5 正态分布的性质

我们在本书的前面部分提到了正态分布具有非常好的统计性质，但是并未解释原因。这里我们给出一个解释：正态分布对线性变换和卷积运算是封闭的（closed）。为了说明这些，我们引进几种记法。

假设一个随机变量 X 服从参数为 μ 和 σ 的正态分布，我们可以将其简记为：

$$X \sim \mathcal{N}(\mu, \sigma)$$

这里记号 ~ 表示服从某种分布，花体字母 $\mathcal{N}$ 表示正态分布。

形如 $X' = aX+b$ 的表达式称为随机变量 X 的一个线性变换，这里 a 和 b 为实数。当 X' 与 X 属于同一分布族时，我们说该分布族对线性变换是封闭的。正态分布就具有这种性质。假如 $X \sim \mathcal{N}(\mu, \sigma^2)$，那么

$$X' \sim \mathcal{N}(a\mu+b, a^2\sigma)$$

正态分布对卷积运算也是封闭的。若 $Z=X+Y$，且 $X\sim\mathcal{N}(\mu_X, \sigma_X^2)$，$Y\sim\mathcal{N}(\mu_Y, \sigma_Y^2)$，那么

$$Z \sim (\mu_X+\mu_Y, \sigma_X^2+\sigma_Y^2)$$

我们之前遇见的那些分布都不能满足这些性质。

习题6-10

假设 $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$，$Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$，$Z=aX+bY$，请计算 Z 的分布函数。

习题6-11

让我们看一下如果两个连续型的分布相加，结果会是什么样。首先从指数分布、正态分布、对数正态分布和帕累托分布中任意挑选两个分布，并调整参数使得这两个分布的均值和方差相近。

将两个分布产生的随机数相加，计算其分布。请利用第 4 章中的方法来测试这两个分布的和是否可以用一个连续型的分布来表示。

6.6 中心极限定理

到目前为止，我们已经知道：

- 如果将一些服从正态分布的数据加起来，得到的和也服从正态分布；
- 如果将一些不服从正态分布的数据加起来，那么得到的结果一般情况下不会服从前面讲过的连续分布。

理论上我们已经证明了：如果将大量服从某种分布的值加起来，所得到的和会收敛到正态分布。

假设随机变量 X 的均值和标准差为 μ 和 σ，那么 n 个随机变量 X 的和渐进地服从 $\mathcal{N}(n\mu, n\sigma^2)$。

上述理论称为*中心极限定理*（Central Limit Theorem），它是统计分析中非常重要的工具。但是这个定理的成立要求满足一些条件。

- 用于求和的数据必须满足独立性。

- 数据必须服从同一个分布（这个要求可以被适当地放松）。
- 产生这些数据分布的均值和方差必须是有限的（所以帕累托分布就不能满足这个条件了）。
- 收敛的速度取决于原来分布的偏度。如果数据服从指数分布，那么这些数据的和将会很快收敛；但如果数据服从对数正态分布，那么收敛速度就没那么快了。

中心极限定理部分解释了为什么正态分布在自然界中广泛存在。绝大多数动物（或者其他生命形式）的特征，如体重，都会受到大量遗传和环境因素的影响，而且这些影响是具有可加性的。我们观测到的这些特征是大量微效因素的加和，所以它们都近似地服从正态分布。

习题6-12

假设 $x_1, \cdots, x_n$ 是服从同一分布的独立数据，且均值 μ 和方差 σ^2 都是有限的，那么样本均值服从什么分布呢？

$$\bar{x} = \frac{1}{n}\sum x_i$$

样本均值的方差会随着 n 的增大发生什么样的变化？提示：可以回想一下 6.5 节的开头。

习题6-13

从指数分布、对数正态分布、帕累托分布中选择一个分布函数，然后产生一组随机数（个数为 2、4 或 8 等），计算它们和的分布。画出分布图看看是否接近正态分布？当随机序列的长度多长时会收敛到正态分布？

习题6-14

如果我们不计算它们的总和，而是改成计算它们的乘积，那么随着项数增多，结果会怎么样？ 提示：看看乘积对数的分布。

6.7 分布函数之间的关系框架

到现在为止，我们已经了解了概率质量函数（PMF）、累积分布函数（CDF）和概率密度函数（PDF）等概念。我们回顾一下之前的这些内容，图 6-1 列出了这些函数之间的关系。

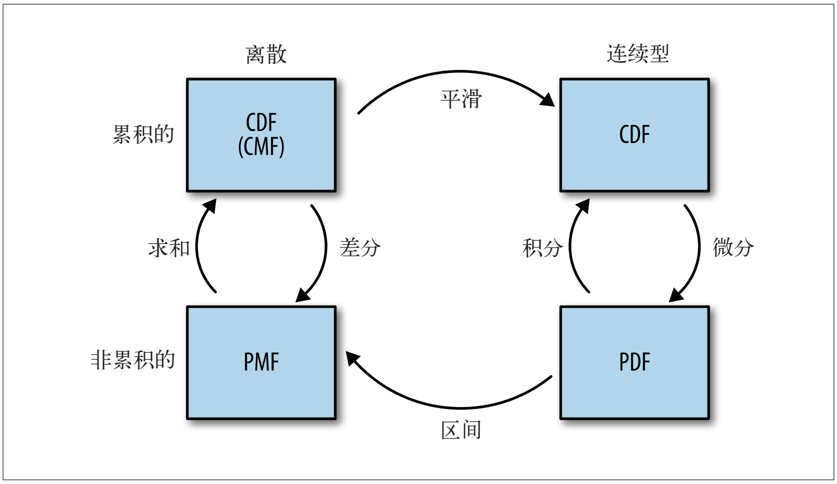


图 6-1：分布函数之间的关系框架

我们从概率质量函数出发，它表示的是离散随机变量在各特定取值上的概率。按随机变量取值的大小将其概率值进行累加，就可以得到累积分布函数。本来为了命名的连贯性，我们应该将这里的累积分布函数称为累积质量函数，但到目前为止，还没有人用过这个词。

我们也可以从累积分布函数出发，通过差分运算得到概率质量函数。

同样地，概率密度函数是连续型累积分布函数的微分，反过来说，累积分布函数是概率密度函数的积分。这里要注意的是，概率密度函数表示的是一个值对应的概率密度而非概率，如果要计算概率，我们必须对概率密度进行积分。

我们可以运用多种方式将一个离散型的分布变成连续型的分布。例如，可以认为这些离散的数据来自一个连续型的分布（比如指数分布

或者正态分布），然后利用这些数据估计分布的参数。这些内容我们将在第 8 章进行更深入地探讨。

假如我们将概率密度函数按一定的区间进行分段，然后在每一段分别进行积分，这样就将一个连续型的分布离散化了，得到了一个近似概率密度函数的概率质量函数。在第 8 章，我们将利用这个方法来做贝叶斯估计。

习题6-15

请编写一个名为 `MakePmfFromCdf` 的函数，用来将分布的 CDF 转换成对应的 PMF。

访问 http://thinkstats.com/Pmf.py 查看本练习的答案。

6.8 术语表

- 中心极限定理（Central Limit Theorem）
  早期的统计学家弗朗西斯·高尔顿爵士认为中心极限定理是“The supreme law of Unreason”。

- 卷积（convolution）
  一种运算，用于计算两个随机变量的和的分布。

- 虚幻的优越性（illusory superiority）
  心理学概念，是指人们普遍存在的将自己高估的一种心理。

- 概率密度函数（probability density function）
  连续型累积分布函数的导数。

- 随机变量（random variable）
  一个能代表一种随机过程的客体。

- 随机数（random variate）
  随机变量的实现。

- 鲁棒性（robust）
  如果一个统计量不容易受到异常值的影响，我们说它是鲁棒的。
- 偏度（skewness）
  分布函数的一种特征，它度量的是分布函数的不对称程度。

第 7 章

假设检验

从 NSFG（美国全国家庭成长调查）提供的数据中，我们可以发现一些很明显的现象，例如第一胎婴儿与非第一胎婴儿相比有很多不同的地方。到目前为止，我们仅从数值大小的角度比较了这些效应（effect）。接下来我们将对这些效应进行统计检验。

最基本的问题是这些效应是真实存在的还是随机引起的。例如，我们发现孕妇第一胎的怀孕周期和非第一 胎的不同，那么这个差异是确实存在的还是偶然引起的呢？

我们难以直接回答上述问题，但可以将其拆成两部分：首先检验这个效应是否具有显著性，然后通过解释统计检验的结果来回答上述问题。

在统计学上，显著性（significant）有专门定义，与通常语义下的用法不同。如本书前面提到的，我们说一个效应在统计学上具有显著性，是指这种情况在一次试验中不大可能（unlikely）发生。

为了让上述表述更加精确，我们必须回答如下三个问题。

1. 什么是“偶然”？

2. 什么是“不大可能发生”？

3. 什么是“效应”？

这三个问题要比看起来难很多，但是人们已经发展出了一套方法来进行统计显著性检验。

- 原假设（null hypothesis）
  基于一种假设的系统模型，在这种假设下我们认为观测到的效应是由偶然因素造成的。

- p 值（p-value）
  在原假设下，出现直观效应的概率。

- 解释（Interpretation）
  基于 p 值的大小，推断观测到的效应是否具有统计显著性。

上述过程称为假设检验（hypothesis testing）。这里潜在的逻辑类似于数学上的反证法：为了证明数学命题 A 是正确的，我们先假设 A 是错误的，如果基于这个假设得出了矛盾的结果，那么我们就证明了 A 是正确的。

同样地，为了检验某个直观效应是否真实存在，我们首先假设这个效应不是真实存在的，即偶然造成的（原假设）。然后基于这个原假设计算出发生这种效应的概率（p 值）。如果 p 值非常小，我们就可以认为原假设不大可能是真的。

7.1 均值差异的检验

统计检验中最简单的一种检验是比较两组数据的均值是否存在差异。在 NSFG 数据中，我们发现第一胎婴儿怀孕周期的均值略长于非第一胎婴儿怀孕周期的均值，同时第一胎婴儿出生体重的均值略轻于非第一胎婴儿出生体重的均值。接下来我们将检验这些差异是否具有统计显著性。

在上述两个例子中，原假设是两个分组的分布相同，出现上述差异是随机因素引起的。

为了计算 p 值，我们把所有婴儿（包括第一胎和非第一胎）的数据混在一起。然后重新随机分成两组：第一组的样本个数等于第一胎婴儿的样本数，第二组的样本个数等于非第一胎婴儿的样本数。每次分完组后，计算两个分组的均值的差。这个差值相当于在原假设（两个分组没有差异）下观测到的差值。

如果我们产生足够数量的这种分组样本，可以统计有多少个差值（由于随机因素引起的）大于等于实际上我们观测到的差值，这个比例就是 p 值。

就怀孕周期而言，我们观察了 n=4413 个第一胎婴儿，m=4735 个非第一胎婴儿。两组样本均值的差值为 δ=0.078 周。为了计算这个效应的 p 值，将两个分组的数据合在一起，然后随机将这些数据分成两组，一组的样本数为 n，另一组为 m，再计算这两个分组的均值的差。

这是重抽样（resampling）的一个例子，因为我们是从一个分布的样本数据里面重新随机地抽取样本。我们随机产生了 1000 次这样的过程，这些差值的分布如图 7-1 所示。

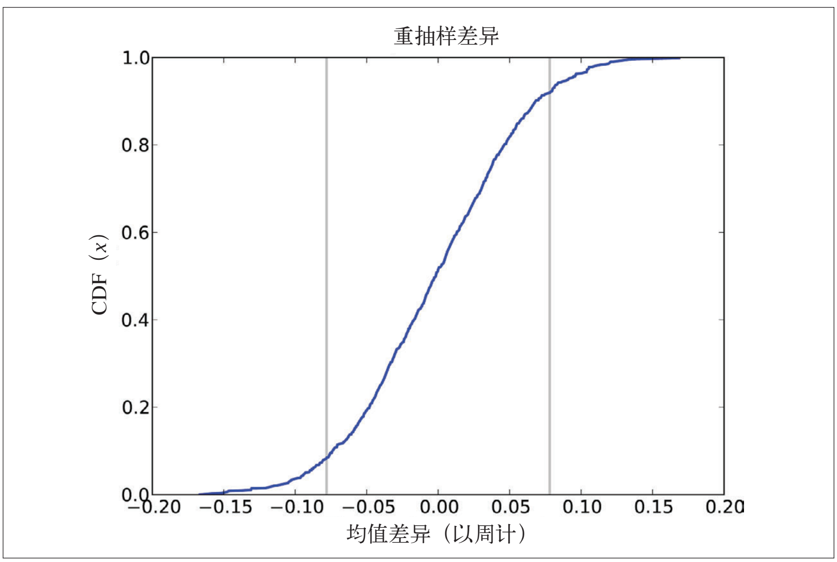


图 7-1：重抽样数据均值差异的 CDF

从图中我们发现差值的平均值很靠近 0，这跟我们的原假设是一致的。图中的两条竖线表示两个阈值（cutoff），这里选择了 $x=-\delta$ 和 $x=\delta$。

在这 1000 个差值里面，我们发现有 166 个值的绝对值大于等于 δ，所以这里的 p 值约为 0.166。换句话说，在两个分组的怀孕周期没有差别的原假设下，出现这种效应的概率大约为 17%。

这样的效应在一次试验中是不大可能出现的，但是这个概率是否足够小呢？我们将在下一节讨论这个问题。

习题7-1

在 NSFG 的数据集中，第一胎婴儿的平均体重与非第一胎婴儿的平均体重的差异为 2.0 盎司。请计算这个差异的 p 值。

提示：这里的重抽样应该用的是有放回抽样。因此应该使用 `random.choice`，而不是 `random.sample`（参见 3.8 节）。

你可以借鉴本节中我用来生成结果的代码，下载代码请访问 http://thinkstats.com/hypothesis.py。

7.2 阈值的选择

在假设检验中，我们必须注意两种类型的错误 。

- I 类错误（type I error），也称假阳性（false positive），指的是我们接受了一个本质为假的假设。也就是说，我们认为某个效应具有统计显著性，但实际上该效应却是由偶然因素产生的。
- II 类错误（type II error），也称假阴性（false negative），指的是我们推翻了一个本质为真的假设。也就是说我们将某个效应归结为随机产生的，但实际上真实存在。

假设检验中最常用的方法是为 p 值选择一个阈值 [1]α，一旦 p 值小于这

注 1：又称为显著性准则（significance criterion）。

个阈值，我们就推翻原假设。通常情况下我们选择 5% 为阈值。在这个标准下，第一胎和非第一胎婴儿怀孕周期的差异就不具备统计显著性了，但是在出生体重上的差异就具有显著性。

对于这类假设检验，我们可以得到出现假阳性的精确概率，这个概率就是 α 值。

我们解释一下原因，首先回顾假阳性和 p 值的定义：假阳性是指接受了一个不成立的假设，p 值是指假设不成立时出现测量效应的概率。

两者结合起来，我们的问题是：如果选择 α 为显著性阈值，当假设不成立时，出现该测量效应的概率会是多少呢？答案就是 α。

我们可以通过降低阈值来控制假阳性。例如如果我们设置阈值为 1%，那么出现假阳性的概率就等于 1%。

但是降低假阳性也是有代价的。阈值的降低会导致判断效应确实存在的标准提高，这样推翻有效假设的可能性就变大，即我们更有可能接受原假设。

一般说来，I 类错误和 II 类错误之间存在一种权衡，同时降低这两种错误的唯一方法是增加样本数量（或者，在某些情况下降低测量误差）。

习题7-2

为了研究样本数量对 p 值的影响，请读者试着去掉一半 NSFG 的数据，再计算一下 p 值，并比较结果。如果去掉 3/4 的数据呢？提示：使用 `radom.sample`。

最少需要多少样本量才能保证差异有 5% 的显著性？如果要求有 1% 的显著性，又需要多少样本？

读者可以从 http://thinkstats.com/hypothesis.py 下载到本节所用到的代码。

7.3 效应的定义

当人们看到不寻常的事情发生时通常会感到稀奇，并问："出现这种情况的可能性有多大呢？"这是因为人们在直觉上认为不大可能发生的事情会很少发生。但这种直觉并不总是能经得住推敲。

例如，假设我抛 10 次硬币，每抛一次我都将结果记录下来，用 H 表示正面朝上，用 T 表示反面朝上。如果得到的结果是 THHTHTTTHH，或许大家不会觉得惊讶。但是如果结果是 HHHHHHHHHH，人们可能会问："这种可能性有多大呢？"

在这个例子中，两个结果序列相同的概率为 1/1024。同样地，任意两个结果序列相同的概率也是 1/1024。所以当我们问"出现这种情况的可能性有多大"的时候，必须明确"这种情况"具体是什么。

在 NSFG 数据中，我们将效应定义为"两个分组的均值差（不分正负）大于等于 δ"。给定这个定义之后，我们只关注这个差值绝对值的大小，而不再关注它是正数还是负数。

上述类型的检验称为*双边检验*（two-side test），我们考虑了图 7-1 中分布的两边的情况（正的和负的）。在这里，双边检验的假设是两个分布的平均值有显著差别，而不关注相对大小。

与双边检验对应的是*单边检验*（one-side test）。单边检验关注的是第一胎婴儿数据的均值是否显著高于非第一胎婴儿数据的均值。因为单边检验的假设更具有特异性，所以单边检验的 p 值会比较低，在这里大约是双边检验 p 值的一半。

7.4 解释统计检验结果

本章开头，我们提到了一个问题：如何确定观测到的表观效应是否真实存在？我们是这样来处理的。首先，定义原假设（效应不存在）为 H_0；然后定义 p 值为 $P(E|H_0)$，这里的 E 表示的是与表观效应相符以及比表观效应更显著的效应。最后我们可以计算得到 p 值，并将其与

阈值 α 作比较。

这些步骤非常重要，但是并没有回答我们原来的问题，即这个效应是否真实存在。所以我们应该对假设检验的结果进行解释。一般说来有如下的几种解释。

- 古典解释

  在古典的假设检验中，如果 p 值小于阈值 α，那么我们可以说效应在统计学上是显著的，但是不能得到效应真实存在的结论。这种解释很谨慎，避免提到结论，但无法让人满意。

- 实际解释

  在实际应用中，人们并没有像上述那样正式地处理假设检验。在绝大多数科学杂志中，研究者毫无节制地报道 p 值，读者也将它们作为表观效应真实存在的证据。p 值越低，就越能使他们相信结论的正确性。

- 贝叶斯统计解释

  实际上我们想知道的是 $P(H_A|E)$，这里 H_A 是与 H_0 相对的假设，即效应是真实存在的。由贝叶斯定理可得

  $$P(H_A\,|\,E) = \frac{P(E\,|\,H_A)P(H_A)}{P(E)}$$

  这里 $P(H_A)$ 是在我们观测到这个效应之前的先验概率。$P(E|H_A)$ 是在 H_A 成立的条件下观测到效应 E 的概率。$P(E)$ 是在任意情况下观测到效应 E 的概率。效应要么存在，要么不存在，所以这里 $P(E)$ 可以表示为

  $$P(E) = P(E|H_A)P(H_A) + P(E|H_0)P(H_0)$$

例如，我们要计算 NSFG 数据中怀孕周期的 $P(H_A|E)$。已经知道 $P(E|H_0)$=0.166，所以接下来要做的就是计算 $P(E|H_A)$，并为 H_A 选择一个先验概率。

为了计算 $P(E|H_A)$，我们假设效应是真实存在的，且两个分组均值的差（等于 0.078）反映的是真实的效应。（这样的处理实际上并不严

谨，下一节会解释如何解决这个问题。）

在两个分组中单独地抽取样品，构建每个分组的分布。重复 1000 次这样的试验，我们得到 $P(E|H_A)$ 的估计为 0.494。假设 $P(H_A)$ 的先验概率为 0.5，得到 H_A 的后验概率为 0.748。

因此，若 $P(H_A)$ 的先验概率为 50%，用观测到的证据更新之后得到的后验概率接近 75%。后验概率高于先验概率，这个结果是有意义的，因为这表明了观测到的数据在一定程度上支持了 H_A。不过这个结果看起来多少有点使人惊讶，先验概率和后验概率会差别这么大，而且还是在两个分组均值差异并不具备统计显著性的情况下。

实际上，这一节中所用的方法并不严谨，上述方法倾向于夸大了观测到的差异的影响。下一节我们修正这一倾向。

习题7-3

在 NSFG 的数据中，第一胎婴儿体重的分布与非第一胎婴儿体重的分布不同的后验概率是多少？

读者可以从 http://thinkstats.com/hypothesis.py 下载本节用到的代码。

7.5 交叉验证

在前一个例子中，我们使用数据集来构建 H_A，然后再用同一个数据集进行检验。这并不是一个好方法，很容易产生错误结果。

这里会出现的问题是：即使原假设是真的，也可能因为随机抽样的缘故而导致两个分组的均值有差别（δ）。如果直接用这个差别计算 H_A，然后再用同样的数据来计算 $P(H_A|E)$，那么即使 H_A 为假，也会得到一个很高的 $P(H_A|E)$。

可以用交叉验证（cross-validation）的方法来解决这个问题：用一批数据来计算 δ，然后再用另一批数据来计算 $P(H_A|E)$。第一批数据称为训练集（training set），第二批数据称为测试集（ testing set）。

在像 NSFG 这类包含不同周期不同人群的研究中，可以利用一个周期的数据做训练集，然后用另一个周期的数据做测试集。或者也可以随机地将数据分成两部分，一部分是训练集，另一部分是测试集。

我们按第二种方法将第 6 周期收集到的数据随机地分成两部分。重复数次之后，得到的 $P(H_A|E)$ 的平均值是 0.621。跟预期一致，观测到的差异对检验的影响变小了。一方面是因为我们所用的样本量变小了，另一方面则是训练集和测试集已经不是同一批数据了。

7.6 报道贝叶斯概率的结果

在上一节中，我们选择 0.5 作为 $P(H_A)$ 的先验概率。对于一组假设，假如我们认为它们的可能性都是一样的，即没有哪个假设比其他假设更可能是真的，那么通常我们会指定同一个先验概率。

贝叶斯概率依赖于先验概率的指定，而人们在这个问题上往往很难达成一致，一些人因此对贝叶斯概率持反对态度。对那些坚持认为科学结果应是具有客观性和普遍性的人来说，贝叶斯概率的这种性质是他们无法接受的。

针对反对观点，下面是一种解释：在实际应用中，强有力的证据会降低先验概率的影响，所以即使人们初始指定的概率不同，最终的后验概率会倾向于收敛。

另一种报道贝叶斯概率的方法是只报道似然比（likelyhood ratio）$P(E|H_A)/P(E|H_0)$，而不再关注后验概率。这样，人们就可以依据自己的观点来设置先验概率和计算后验概率。似然比有时也称为贝叶斯因子（Bayes factor，参考 http://wikipedia.org/wiki/Bayes_factor）。

习题7-4

假设 H_A 的先验概率为 0.3，之后我们观测到了新的证据 E，并知道此时似然比 $P(E|H_A)/P(E|H_0)$ 为 3，那么这种情况下 H_A 的后验概率是多少？

习题7-5

下面的例子来自 MacKay 的 *Information Theory, Inference, and Learning Algorithms*：

> 有两个人在犯罪现场留下了血样。奥利弗是其中的一个嫌疑犯，他的血型为O型。犯罪现场的两份血液组织分别是O型和AB型，前者在当地人群中有广泛分布，60%的人是这种血型，后者只有1%。那么这些证据（即留在犯罪现场的血样）是否会增加我们对奥利弗是嫌疑犯的怀疑？

提示：计算证据的似然比，如果大于 1 就说明证据支持对奥利弗的怀疑。读者可从 MacKay 一书的第 55 页找到问题的答案和解释。

7.7 卡方检验

7.2 节中我们得出了以下结论：第一胎婴儿的平均怀孕周期与非第一胎婴儿的平均怀孕周期的差别不具备统计显著性。但在 2.10 节我们计算相对风险时，发现第一胎婴儿倾向于更早或者更晚出生，而较少准时出生。

综上，或许这两个分布有相同的均值，但却有不同的方差。我们本来应该检验方差的差异是否具有显著性，但方差相对均值而言鲁棒性较差，针对方差的统计检验通常表现较差。

这里我们采取的方法是直接检验这种趋势是否具有统计学意义上的差别，即第一胎婴儿倾向于更早或者更晚出生而较少准时出生，这种差异具有统计学意义。

我们分五步来完成这个检验。

1. 按 2.10 节中的标准把数据按怀孕周期分成三个分组（提前出生、准时出生和延后出生）。因为我们有两组数据，所以总共有 6 个单元格（cell）。

2. 计算每个单元格期望出现的数字。在原假设下两个分组是相同的，所以我们将两组数据混在一起，来估计 P(提前出生)、P(准时出生) 和 P(延后出生)。

   我们有 n=4413 个第一胎婴儿的数据，在原假设下，我们期望会有 nP(提前出生) 个婴儿提前出生，nP(准时出生) 个婴儿准时出生，等等。同样地，对 m=4735 个非第一胎婴儿样本，我们可以计算出每个单元格的期望数值。

3. 对每个单元格，计算观测到的数值（O_i）与期望数值（E_i）的离差，即 O_i-E_i。

4. 计算某种形式的离差和，将这个量称为*检验的统计量*。通常我们会选择卡方统计量：

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

5. 利用蒙特卡罗模拟来计算 p 值，这个 p 值表示的是在原假设下出现比观测值（即我们在第 4 步中计算得到的统计量的值）更高的卡方统计量的概率。

当检验中用到的统计量是卡方统计量时，我们称该统计检验为卡方检验（chi-square test）。卡方统计量服从卡方分布，据此我们可直接计算出统计检验的 p 值。

在 NSFG 的数据中，我们计算得到 χ^2=91.64, p 值小于 0.0001。这样我们可以认为该结果具有统计显著性。需要注意到的一点是：我们仍然在用同一批数据进行检验，最好能在另一批数据上验证一下我们的结果。

读者可以从 http://thinkstats.com/chi.py 下载到本节所用的代码。

习题7-6

假设你是一家赌场的老板，你怀疑有个赌客对骰子做了手脚。你已经将赌客抓了起来并且没收了他的骰子。现在你必须证明他的骰子是有问题的。

你掷了 60 次骰子，记录结果如下所示：

点数	1	2	3	4	5	6
频数	8	9	19	6	8	10

用上述结果计算的卡方统计量等于多少？在骰子没有问题的情况下，卡方统计量比这个值更大的可能性是多少？

7.8 高效再抽样

如果读者之前学习过概率统计的知识，在看到图 7-1 时或许会觉得有点儿不屑一顾，因为我们花费了大量计算机资源去模拟一些本可以用很简单的理论分析就能得到的结果。

很显然，本书并没有把重心放在数学分析上面。我们更愿意通过计算机用一些貌似“愚蠢”的方法来讲述本书的内容，这样可能更容易让初学者明白这里的意义，也更容易让他们上手。所以只要我们的模拟不需要耗费很长时间，这种方式没有什么不妥。

但有时候，只需进行一些简单的分析就可以节省大量的计算，图 7-1 就是这样的一个例子。

我们之前是将两个分组的怀孕周期混在一起，然后再按原来每组的个数重新随机将数据分成两组，之后计算两个分组均值的差异。重复这样的过程，用这些数据构建分布。

这里，我们可以通过分析的方法直接计算均值差值的分布。假设怀孕周期服从一个分布，均值为 μ，方差为 σ^2。我们从这个分布随机抽取 n 个样本，那么根据中心极限定理，样本的和渐进服从 $\mathcal{N}(n\mu, n\sigma^2)$。

为了得到样本均值的分布，这里需要用到正态分布的一个性质：若 X 服从正态分布 $\mathcal{N}(\mu, \sigma^2)$，那么

$$aX+b \sim \mathcal{N}(a\mu+b, a^2\sigma^2)$$

设 $a=1/n$，$b=0$，等式两边同除以 n 则

$$X/n\sim\mathcal{N}(\mu/n, \sigma^2/n^2)$$

所以样本均值渐进服从 $\mathcal{N}(\mu, \sigma^2/n)$。

为了得到样本均值差值的分布，我们需要用到正态分布的另一个性质：若 X_1 服从 $\mathcal{N}(\mu_1, \sigma^2_1)$，$X_2$ 服从 $\mathcal{N}(\mu_2, \sigma^2_2)$，那么

$$aX_1+bX_2\sim(a\mu_1+b\mu_2, a^2\sigma^2_1+b^2\sigma^2_2)$$

这种情况下有：

$$X_1-X_2\sim(\mu_1-\mu_2, \sigma^2_1+\sigma^2_2)$$

综上，可知图 7-1 的分布服从 $\mathcal{N}(0, f\sigma^2)$，这里 $f=1/n+1/m$。将 $n=4413$ 和 $m=4735$ 代入公式，得到最近的分布为 $\mathcal{N}(0, 0.0032)$。

我们用 `erf.NormalCdf` 来计算均值差值的 p 值：

```
delta = 0.078
sigma = math.sqrt(0.0032)
left = erf.NormalCdf(-delta, 0.0, sigma)
right = 1 - erf.NormalCdf(delta, 0.0, sigma)
```

计算得到的双边检验的 p 值为 0.168，非常接近我们用重抽样的方法计算得到的结果 0.166。

读者可从 http://thinkstats.com/hypothesis_analytic.py 下载本节所用到的代码。

7.9 功效

当检验的结果为阴性时（即检验结果不具备统计显著性），我们是不是就可以认为所要检验的效应不存在呢？这其实还依赖于所用的统计检验的功效。

统计功效（statistical power）指的是在原假设为假的情况下，检验的

结果为阳性的概率。一般地，一个统计检验的功效依赖于样本数量、效应的大小和我们设置的阈值 α。

习题7-7

7.2 节中，在 α=0.05 并假设两个分组真实的差异等于 0.078 周的情况下，这个检验的功效是多少？ 如果 α=0.1 呢？

可以通过从两个分布中生成随机样本（满足均值差异为 0.078）来估计功效：对这些数据进行检验，阳性结果比例就是这种检验的功效。

当检验结果为阴性时，一种报道检验统计功效的方式是：如果出现的效应为 x，那么这种检验推翻原假设的概率为 p（即功效）。

7.10 术语

- 单元格（cell）
  在卡方检验中，将观测按一定的标准分到各个单元格里，每个单元格代表一种分类。

- 卡方检验（chi-square test）
  用卡方统计量做统计量的统计检验 。

- 交叉验证（cross-validation）
  交叉验证使用一个数据集进行探索性数据分析，然后用另一个数据集进行测试。

- 假阴性（false negative）
  在效应真实存在的情况下，我们认为这个效应是由偶然因素引起的。

- 假阳性（false positive）
  在原假设为真的情况下，我们拒绝了原假设的结论。

- 假设检验（hypothesis testing）
  判定出现的效应是否具有统计显著性的过程。

- 似然比（likelihood ratio）
  一种概率的比值，$P(E|A)/P(E|B)$，这里 A 和 B 是两种假设。似然比不依赖于先验概率，可以用来报道贝叶斯统计推断的结果。

- 原假设（null hypothesis）
  一种基于以下假设的模型系统：我们观测到的效应只是由偶然因素引起的。

- 单边检验（one-sided test）
  一种检验类型，关注的是出现比观测到的效应更大（或小）的效应的概率。

- p 值（p-value）
  在原假设成立的情况下，出现我们观测到的效应的概率。

- 功效（power）
  在原假设为假的情况下，检验推翻原假设的概率。

- 显著性（significant）
  我们说某个效应具有统计显著性指的是这种情况不大可能是由偶然因素引起的。

- 检验统计量（test statistic）
  衡量观测到的效应与原假设下期望的结果之间偏差的统计量。

- 测试集（testing set）
  用做测试的数据集。

- 训练集（training set）
  用做训练的数据集。

- 双边检验（two-sided test）
  一种检验类型，关注的是出现比观测到的效应更大的效应的概率，不考虑正负。

第 8 章

估计

8.1 关于估计的游戏

让我们从一个游戏开始。我想到一个分布，然后让你去猜这个是什么分布。我们将从简单的情况出发，逐步开始我们的讨论。

关于我心里想到的那个分布，我会提示两点：一是这是一个正态分布，二是我们有一组从这个分布中得到的随机样本：

{−0.441, 1.774, −0.101, −1.138, 2.975, −2.138}

你认为这个分布的均值参数 μ 会是多少呢？

一种简单的方法是用样本的均值去估计 μ。到目前为止，我们一直用 μ 这个符号表示样本均值和分布的均值。但从现在开始，我们将区分这两个概念。我们用 $\bar{x}$ 表示样本均值。在这个例子中，$\bar{x}$ =0.155，所以我们有理由猜测 μ=0.155。

上述过程称为*估计*（estimation），用来估计分布参数的统计量（在这里是样本均值）称为*估计量*（estimator）。

利用样本均值来估值 μ 似乎无可厚非，我们很难再想到有什么比它更

好的估计量了。接下来假设我们改变了游戏规则，在随机样本中加入一些异常值。

现在游戏变成这样：我想到一个分布，然后告诉猜的人这是个正态分布。但是随机样本是由一个粗心的人来抽取，他有时会把小数点标在错误的位置，于是最终得到了这样的一组数据：

$\{-0.441, 1.774, -0.101, -1.138, 2.975, -213.8\}$

那么这时对 μ 的估计应该是多少呢？假如我们还是用样本均值，那么估计的结果是 −35.12。这是最好的估计结果吗？有没有其他方法可以用来估计 μ 呢？

一种直观的方法是先鉴定出异常值并对它们进行修剪，然后再用剩下的样本的均值来估计参数。除此之外，还有一种方法是用中位数作为估计量，而不再用样本均值。

选择哪一种方法要视具体情况（例如，是否存在异常值）和估计的目的而定，是要让误差最小，还是要让得出正确答案的可能性最大？

假设不存在异常值，那么样本均值会最小化均方误差（Mean Squared Error，MSE）。假设我们多次进行这个游戏，每次游戏结束后计算 $\bar{x}-\mu$，样本均值会使得下式达到最小值：

$$\text{MSE}=\frac{1}{m}\sum(\bar{x}-\mu)^2$$

这里，m 表示的是游戏进行的次数（这里不要同 n 混淆了，n 表示的是每次游戏中得到的样本的数量）。

用样本均值估计 μ 能最小化均方误差，这是一个非常好的性质，但它并不总是最优策略。例如：假设我们正在评估一个建筑工地的风速分布，如果估计得太高，我们可能会建造过多的结构，导致成本增加；但如果估计得过低，大楼可能会倒塌。这时误差的损失函数并不是对称的，最小化均方误差就不是最优策略了。

另一个例子，假设我掷了三次六面骰子然后问你三次得到的点数总和

是多少，如果你猜对了会得到一个奖品，猜错了则什么也没有。这种情况下，如果我们要让均方误差最小，得到的结果是 10.5，显然这是个糟糕的数字。这里，你需要的是一个能使得你有最大的概率猜对的估计，也即*极大似然估计量*（Maximum Likelihood Estimator，MLE）。如果你选择 10 或 11，那么你将有 1/8 的机会猜对，这个才是你的最优选择。

习题8-1

编写一个函数，从一个均值为 0、方差为 1 的正态分布中产生 6 个随机数，利用这些随机数估计均值，并计算误差 $\bar{x}-\mu$。运行 1000 次这个函数，计算均方误差。

接下来修改一下函数，改为用中位数作为均值的估计，再计算此时估计的均方误差。请读者试着比较这两种估计的差别。

8.2 方差估计

我们依然从上一节那个游戏出发，现在我想到的是个正态分布，也得到了一批样本：

{−0.441, 1.774, −0.101, −1.138, 2.975, −2.138}

那么这个分布的方差是多少？同样地，直观的想法是用样本方差来估计分布的方差。我们用 S^2 来表示样本方差，并将其同分布方差 σ^2 区分开。

$$S^2=\frac{1}{n}\sum(x_i-\bar{x})^2$$

在样本数量足够多的情况下，S^2 是一个很好的估计量；但是如果样本数量很少，那么 S^2 会低估 σ^2。因为这个不幸的性质，S^2 只是 σ^2 的一个*有偏*估计。

如果在进行很多次游戏之后，我们发现估计量与真实参数的误差的平均值为 0，那么我们就称这个估计量是*无偏的*（unbiased）。σ^2 的一个无偏估计 S^2_{n-1} 是：

$$S_{n-1}^2 = \frac{1}{n-1}\sum(x_i - \bar{x})^2$$

这里有个麻烦是“样本均值”可以是 S^2，也可以是 S_{n-1}^2，并没有区分，有时会带来混乱。

http://wikipedia.org/wiki/Bias_of_an_estimator 解释了为什么 S^2 是有偏的，同时证明了 S_{n-1}^2 的无偏性。

习题8-2

编写一个函数，从一个均值为 0、方差为 1 的正态分布中产生 6 个随机数，利用样本方差去估计 σ^2，并计算估计误差 $S^2-\sigma^2$。运行这个函数 1000 次，计算平均的误差（这里没有对误差进行平方）。

接下来修改一下函数，用无偏估计量 S_{n-1}^2 来估计方差，并计算估计的平均误差。当模拟次数增加，估计的平均误差是否收敛到 0 ？

8.3 误差

在继续讨论之前，我们需要理清一些概念。均方差和有偏性都是长期概念，是在多次进行试验后得到的结果。

在进行游戏的过程中，我们不可能知道估计的误差。假设我只给你一个样本数据，然后让你来估计分布的参数。参数可以估计出来，误差却无从得知。如果你能知道误差是多少，那么我们实际上就不用这个估计量了。

这里讨论估计误差是为了描述不同的估计量在多次试验下的表现。本章的目的也是通过一系列的试验来检验这些表现。由于我们的试验是人工设置的，所有的参数都是已知的，所以我们可以计算出误差。但在实际的数据中，我们并不知道参数的真实情况，所以无法得到误差值。

接下来回到我们的讨论。

8.4 指数分布

还是那个游戏，但现在我想到的是指数分布，并且得到了一组样本：

{5.384, 4.493, 19.198, 2.790, 6.122, 12.844}

那么这里指数分布的参数 λ 会是多少呢？

因为指数分布的均值为 $1/\lambda$，所以根据之前的处理，我们会选择如下的估计量来估计 λ：

$$\hat{\lambda} = 1/\bar{x}$$

一般情况下，我们给待估的参数上加一个帽子，以此来表示这个参数的估计量。这里 $\hat{\lambda}$ 不仅是 λ 的估计量而且还是它的极大似然估计量，[1] 所以如果我们想要有最大的可能性猜对 λ，$\hat{\lambda}$ 是最好的选择。

但是我们出现异常值时，$\bar{x}$ 的鲁棒性不好，所以 $\hat{\lambda}$ 也会面临同样的问题。

这里可以用另一种基于中位数的方法来估计 λ。我们知道指数分布的中位数等于 $\log(2)/\lambda$，于是根据前面的做法，定义 λ 的估计量为：

$$\hat{\lambda}_{1/2} = \log(2)/\mu_{1/2}$$

这里 $\mu_{1/2}$ 表示的是样本的中位数。

习题8-3

请通过模拟比较一下 $\hat{\lambda}$ 和 $\hat{\lambda}_{1/2}$ 的均方误差哪个更低，并检查这两个估计量是否是有偏的。

8.5 置信区间

到目前为止，我们学习了用估计量产生一个值来估计参数，这种方法称为点估计（point estimation）。在很多问题中，有时我们更希望知道

注 1：参见 http://wikipedia.org/wiki/Exponential_distribution#Maximum_likelihood。

一个有上界和下界的区间，这个区间能够覆盖未知的参数。

更一般地，我们想知道整个分布的情况，也就是分布参数所有取值的范围，在此范围内的每一个值，以及每个值的可能性。

我们从置信区间（confidence interval）这个概念开始。

回到我们之前的游戏。我想到一个指数分布，然后告诉你一组样本：

{5.384, 4.493, 19.198, 2.790, 6.122, 12.844}

接下来，我想让你给我一个范围，这个范围有很大的可能性覆盖未知参数 λ。更具体地说，我想要一个 90% 的置信区间，也就是如果我重复地进行这个游戏，平均而言这个区间能 90% 包含 λ。

这样的游戏有点太难了，所以这里直接给出了答案，读者可以试着去验证一下结果。

我们通常以缺失率（miss rate）α 来描述置信区间，90% 的置信区间对应的 α=0.1。指数分布的参数 λ 的置信区间为：

$$\left(\hat{\lambda}\frac{\chi^2(2n,1-\alpha/2)}{2n},\ \hat{\lambda}\frac{\chi^2(2n,\alpha/2)}{2n}\right)$$

这里 n 表示样本数量，$\hat{\lambda}$ 是上一节提到的参数的均值估计。$\chi^2(k, x)$ 是自由度为 k 的卡方分布的累积分布函数在 x 处的值（卡方分布请参 http://wikipedia.org/wiki/Chi-square_distribution）。

一般说来，很难用分析的方法推导出参数的置信区间，但用模拟的形式来估计它相对容易很多。接下来我们从贝叶斯统计的角度来讨论参数的估计。

8.6 贝叶斯估计

假如你收集了一些样本，然后根据这些样本计算出了参数的 90% 的置信区间，这似乎意味着参数有 90% 的可能性落在这个区间内。但如果从频率学派的角度来看，这种观点是错误的，因为他们认为虽然分布

的参数未知，但它是一个固定的数字，并不是一个随机变量，参数要么在这个区间内，要么不在。这样频率学派关于概率的观点在这里就不适用了。

那么让我们改变一下游戏的规则。

我想到一个分布：这是一个指数分布，但我是从一个 (0.5, 1.5) 的均匀分布中随机抽了一个值作为参数 λ。下面是抽样得到的一组样本，我们用 X 表示：

{2.675, 0.198, 1.152, 0.787, 2.717, 4.269}

基于这组样本，你觉得 λ 的值会是多少？

在这个版本的游戏中，λ 是一个随机变量，所以我们完全可以讨论它服从什么样的分布，并且使用贝叶斯定理很容易就能计算出结果。

下面是计算步骤。

1. 将 (0.5, 1.5) 划分成一组长度相等的小区间。对每个小区间，我们定义假设 H_i 为：λ 落在第 i 个区间。因为 λ 服从均匀分布，所以 H_i 的先验概率 $P(H_i)$ 对所有的 i 都是相等的。

2. 对每一个假设 H_i，我们计算似然函数 $P(X|H_i)$，即在 H_i 的条件下出现样本 X 的概率[2]：

$$P(X \mid H_i) = \prod_j \mathrm{expo}(\lambda_i, x_j)$$

这里 $\mathrm{expo}(\lambda, x)$ 表示的是参数为 λ 的指数分布在 x 处的概率密度函数。

$$\mathrm{PDF}_{\mathrm{expo}}(\lambda, x)=\lambda \mathrm{e}^{-\lambda x}$$

符号 Π 的意思请参考 http://wikipedia.org/wiki/Multiplication#Capital_Pi_notation。

注 2：这里对 H_i 进行了离散化处理，我们选择第 i 个小区间中的一个值 x_i 来代替 H_i，因为指数分布的参数是一个数。这样只要我们划分的区间数量足够多，近似处理后的结果与用积分方式计算的结果就很接近。——译者注

3. 然后利用贝叶斯定理计算后验概率

$$P(H_i|X)=P(H_i)P(X|H_i)/f$$

f 是一个归一化因子

$$f=\sum_i P(H_i)P(X|H_i)$$

得到了参数的后验分布后，就很容易计算置信区间了。例如，90% 的置信区间的上下限就可以选择后验分布 95% 和 5% 的百分位数。

贝叶斯置信区间有时又称为可信区间（credible interval）。贝叶斯统计定义的置信区间与频率学派定义的置信区间之间的差别请参考 http://wikipedia.org/wiki/Credible_interval。

8.7 贝叶斯估计的实现

我们可以用 Pmf 和 Cdf 等表示分布的函数来表示先验分布，但是因为我们要将假设映射为概率，所以 Pmf 是更好的选择。

Pmf 的每一个取值都代表了一个假设，例如，0.5 表示假设 λ=0.5。在先验分布中，所有假设的概率相等。所以我们可以这样构造先验分布：

```
def MakeUniformSuite(low, high, steps):
    hypos = [low + (high-low) * i / (steps-1.0) for i in
range(steps)]
    pmf = Pmf.MakePmfFromList(hypos)
    return pmf
```

这个函数生成了 λ 先验的 Pmf，Pmf 所有取值构成了我们所有的假设，我们把这个假设的集合称为一个 suite。所有假设有相同的概率，所以这个 Pmf 是一个均匀分布。

参数 low 和 high 定义了 λ 取值的范围，steps 定义了假设的数量。

为了更新所有 H_i 的概率，我们输入先验 Pmf 和所有的观测数据（即所谓的“证据”）：

```
def Update(suite, evidence):
    for hypo in suite.Values():
        likelihood = Likelihood(evidence, hypo)
        suite.Mult(hypo, likelihood)
    suite.Normalize()
```

对 suite 中的每一个假设，将假设的先验概率乘以在这个假设下的似然值，然后再对 suite 进行归一化。

在这个函数中，suite 必须是一个 Pmf，evidence 数据可以是任意形式，只要 Likelihood 函数能够识别就行。

Likelihood 函数的定义为：

```
def Likelihood(evidence, hypo):
    param = hypo
    likelihood = 1
    for x in evidence:
        likelihood *= ExpoPdf(x, param)

    return likelihood
```

在 Likelihood 中，我们将 evidence 当成是一组来自指数分布的样本，它可以计算上一节的 Π。

ExpoPdf 计算指数分布在 x 处的概率密度：

```
def ExpoPdf(x, param):
    p = param * math.exp(-param * x)
    return p
```

将这些代码放在一起，我们就可以构造 λ 先验分布，并且计算其后验分布：

```
evidence = [2.675, 0.198, 1.152, 0.787, 2.717, 4.269]
prior = MakeUniformSuite(0.5, 1.5, 100)
posterior = prior.Copy()
Update(posterior, evidence)
```

读者可以从这里下载本节用到的代码 http://thinkstats.com/estimate.py。

在考虑贝叶斯估计的时候，我会想象有满满一屋子的人在猜测一个我想要估计的值，在这个例子里这个值就是 λ。起初，屋里的每一个人

都对自己猜测的值有一个置信度（即相信有多大的可能性猜对）。

当观测到新的证据后，所有人都根据 $P(E|H)$ 对自己的置信度进行更新，这里 $P(E|H)$ 指的是认为起初猜测正确的假设下，E 的似然值。

通常似然函数会计算出概率，其最大值为 1。因此，一开始每个人的置信度通常会下降（或保持不变），但当我们对结果进行归一化后，每个人的置信度又会上升。

所以，上述过程的净效应就是有些人的置信度上升了，有些人的下降了。而这取决于他们假设的相对似然值。

8.8 删失数据

下面的问题来自 David MacKay 的 *Information Theory, Inference and Learning Algorithms* 一书的第 3 章。具体内容读者可从这里下载 http://www.inference.phy.cam.ac.uk/mackay/itprnn/ps/。

> 一个不稳定的粒子从放射源射出，粒子衰变的距离为x。理论上，x服从一个参数为λ的指数分布。实际中，衰变仅能在一个长度从1 cm到20 cm的窗口内观测到。假设我们观测到了n次衰变，衰变的位置为 $\{x_1, \cdots, x_N\}$，那么λ是多少？

这是一个有删失数据（censored data）的估计问题，即有些数据被系统性地排除在外了。

贝叶斯估计一个很大的优势是它可以相对容易地处理删失数据。只要稍微改一下上一节用到的方法就可以处理这个例子中的问题，我们将 $\mathrm{PDF}_{\mathrm{expo}}$ 替换成 $\mathrm{PDF}_{\mathrm{cond}}$：

$$\mathrm{PDF}_{\mathrm{cond}}(\lambda, x) = \lambda \mathrm{e}^{-\lambda x} / Z(\lambda)$$

这里 $1<x<20$，其他情况表达式的值为 0，并且

$$Z(\lambda) = \int_1^{20} \lambda \mathrm{e}^{-\lambda x} \mathrm{d}x = \mathrm{e}^{-\lambda} - \mathrm{e}^{-20\lambda}$$

这里的 $Z(\lambda)$ 在习题 6-5 中出现过。

习题8-4

请下载 http://thinkstats.com/estimate.py，并将下载文件命名为 decay.py，这个文件包含了本章前几节所用到的代码。

请修改 decay.py，然后计算在得到观测 X={1.5, 2, 3, 4, 5, 12} 后 λ 的后验分布。这里 λ 的先验分布可以选择 0 到 1.5 之间的均匀分布（不包含 0）。

读者可以从 http://thinkstats.com/decay.py 下载到问题的一种解答。

习题8-5

在 2008 年明尼苏达州的参议员选举中，Al Franken 得到了 1 212 629 张选票，Norm Coleman 得到了 1 212 317 张选票。Franken 被宣布胜选。但是 Charles Seife 却指出这次选举结果是无效的，因为票数差异的幅度远小于误差的幅度，所以投票结果应该是两位竞选人打成平手。

假设在登记选票时，有可能会漏记选票，也有可能会将同一张选票登记两次，那么 Franken 真正赢得选举的概率是多大?

提示：这里必须添加一些细节来完成建模过程。

8.9 火车头问题

火车头问题是一个非常经典的估计问题，又叫“德国坦克问题”。下面是 Mosteller 在 *Fifty Challenging Problems in Probability* 中提到的版本：

> 铁路公司将它所有的火车头都进行了编号，从1到N。有一天你看见一个编号为60的火车头，那该铁路公司总共有多少火车头呢?

在接着往下讨论之前，我们先想想如下问题。

1. 对于一个给定的估计 $\hat{N}$，观测的似然函数是什么？极大似然估计量是什么？

2. 假如我们看到编号为 i 的火车，我们有理由猜测一个乘数 a，并用 $\hat{N}=ai$ 来估计总的火车数。那么我们该怎么样选择 a 使得估计的结果能使均方误差最小？

3. 仍然假设 $\hat{N}=ai$，我们能找到一个使得 $\hat{N}$ 为无偏估计量的 a 吗？

4. N 是多少时 60 会是观测到的平均值？

5. 在假设先验概率为 1 到 200 上的离散均匀分布的条件下，贝叶斯后验分布是什么样子的？

对于一个给定的估计量 $\hat{N}$，观测到编号为 i（$i \leqslant \hat{N}$）的火车的概率为 $1/\hat{N}$，$i > \hat{N}$ 的概率为 0。所以 N 的极大似然估计量是 $\hat{N} = i$。换言之，如果观测到的火车的编号是 60，而且我们要以最大的概率保证结果的正确性，那么我们就会猜测铁路公司有 60 辆火车。

但是极大似然估计的结果从均方误差的角度来看并不理想。我们可以通过选择一个 $\hat{N} = ai$ 使得估计量的均方误差达到最小，这里唯一需要做的是估计好 a 的值。

假设实际会有 N 辆火车。我们看到编号为 i 的火车后就猜测铁路公司有 ai 辆火车，那么平方误差为 $(ai-N)^2$。

假设我们观测了 N 次，且看到了所有的火车，那么均方差就是：

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(ai - N)^2$$

为了使均方差最小，我们对 a 进行求导：

$$\frac{\text{dMSE}}{\text{da}} = \frac{1}{N}\sum_{i=1}^{N}2i(ai - N) = 0$$

解得：

$$a = \frac{3N}{2N+1}$$

粗略看来这个结果似乎没什么用，因为等式的右边有 N。我们要想知道 a 就必须知道 N，但是如果我们知道了 N，上面的估计就全然不需要了。

考虑到 N 很大的时候 a 收敛于 3/2，所以这里选择对 N 的估计量为 $\hat{N}=3i/2$。

再从无偏估计的角度出发，首先计算平均误差：

$$\mathrm{ME}=\frac{1}{N}\sum_{i=1}^{N}(ai-N)$$

令 ME=0，得到

$$a=\frac{2N}{N-1}$$

当 N 很大时，a 收敛到 2，所以这里我们又可以选择 $\hat{N}=2i$。

到目前为止，我们已经构造了 3 个估计量：i、$3i/2$ 和 $2i$，分别满足极大似然、最小均方误差和无偏性。

我们还有另外一种估计的方式：选择满足总体均值等于样本均值的 $\hat{N}$ 作为 N 的估计量。假设我们观测到一辆火车的编号为 i，样本均值恰好也为 i，这时满足群体的均 $\hat{N}$ 值等于样本均值的火车数的估计量为 $\hat{N}=2i-1$。

最后，我们从贝叶斯统计的角度来回答这个问题。我们计算

$$P(H_n\mid i)=\frac{P(i\mid H_n)P(H_n)}{P(i)}$$

这里 H_n 是一个假设，假设总共有 n 辆火车。i 表示我们的观测，即观测到的火车的编号。当 $i<n$ 时，$P(i|H_n)=1/n$，其余情况为 0，$P(i)$ 是归一化常数。

如果 N 的先验分布是 1 到 200 上的一个均匀分布，我们就遍历这 200 种假设，并计算每种假设的后验概率。读者可以从 http://thinkstats.com/locomotive.py 下载到实现所用的代码。最终的结果如图 8-1 所示。

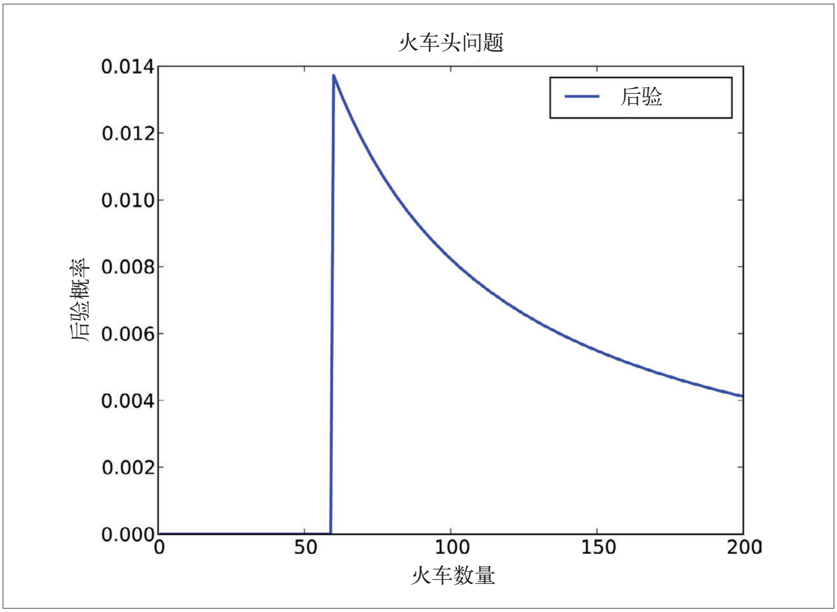


图 8-1：火车数后验分布

根据贝叶斯后验概率，我们得到 90% 的可信区间为 [63, 189]，这仍然是一个非常大的范围。仅仅观测到一辆火车的编号并不能为任意假设提供非常强的证据，虽然它将 $n<i$ 的可能性排除在外了。

如果一开始我们设定不同的先验概率，后验概率就会显著不同，这能帮助你理解为什么其他估计量有多个。

本节我们针对同一个参数构造了多个不同的估计量，我们可以认为这些是从一些不够精确的先验出发得到的结果。假如有足够的先验信息，那么这些估计量将倾向于收敛到同一个值。总之，在我们的例子里，没有一个统计量能够满足我们期望的所有性质。

习题8-6

推广 locomotive.py，使之能处理当观测到的火车数多于 1 辆时的情况，读者只需改动几行代码就可以完成。

看看你能否回答上述情况下的其他问题。维基百科页面 http://wikipedia.org/wiki/German_tank_problem 提供了更多的问题和讨论。

8.10 术语

- 有偏性（bias）
  在平均多次试验的结果后，一个估计量倾向于高估或者低估真实的参数值。
- 删失数据（censored data）
  一种数据集，数据来源于某种采集方式，但是这种采集方式会系统性地排除某些数据。
- 置信区间（confidence interval）
  一种参数的区间估计，以一定的概率包含待估计的参数。
- 可信区间（credible interval）
  贝叶斯统计理论中的置信区间。
- 估计（estimation）
  用样本信息估计分布中未知参数的过程。
- 估计量（estimator）
  用于估计参数的统计量。
- 极大似然估计量（maximum likelihood estimator）
  使得似然函数最大化的估计。
- 均方误差（mean squared error）
  一种衡量估计误差的值。
- 点估计（point estimate）
  用单一的值估计某个参数。

第9章
相关性

9.1 标准分数

本章我们将开始关注变量与变量之间的关系。例如，我们会觉得一般而言身高越高的人体重越重。相关（correlation）就是用来描述这种类型的关系的。

在衡量相关关系的时候会出现的一个问题是，两个变量有不同的度量衡。如身高是用厘米度量的，而体重则是用千克衡量的。还有一个问题，即使两个变量有相同的单位，它们的分布也不同。

有两种方法可以解决这些问题。

1. 将所有的值转换成标准分数（standard score），这就引出了皮尔逊相关系数。
2. 将所有的值转换成百分等级，这就引出了斯皮尔曼相关系数。

假设 X 是一个序列，x_i 是其中的一个值，我们定义标准分数的转换公式为 $z_i=(x_i-\mu)/\sigma$，这里 μ 表示序列的均值，σ 表示标准差。

转换公式的分子表示一个离差，是 x_i 与均值的差异。除以 σ 是为了归一化偏差。这样 Z 的单位就为 1，而且均值为 0，方差为 1。

Z 的分布形状与 X 相似，即如果 X 是一个正态分布，那么 Z 也是一个正态分布；如果 X 的分布函数非对称，或者有一些异常值，那么 Z 也是如此。这类情况下，百分等级转换会提供更为鲁棒的结果。如 R 是 X 的一个百分等级转换结果，那么不论 X 服从什么类型的分布，R 都服从 0 到 100 上的均匀分布（R 的单位为 %）。

9.2 协方差

协方差（covariance）可以用来衡量相关变量变化趋势是否相同。假设我们有两列序列 X 和 Y，它们与其均值离差为：

$$dx_i = x_i - \mu_X$$

$$dy_i = y_i - \mu_Y$$

这里 μ_X 是 X 的均值，μ_Y 是 Y 的均值。如果 X 和 Y 的变化方向一致，那么它们与均值的离差应有相同的正负号。

如果我们将二者的离差相乘，那么当二者的符号相同时，乘积为正数。所以这些乘积加和的结果可以用来衡量两个序列变化是否一致。

协方差就是这些乘积结果的平均值：

$$\mathrm{Cov}(X,Y) = \frac{1}{n}\sum dx_i dy_i$$

这里 n 表示序列的长度（X 和 Y 必须有相同的长度）。

协方差的计算比较简单，但我们一般较少使用，因为这个值很难解释。另一个问题是，协方差的单位是 X 和 Y 的单位的乘积。在前面的那个例子里，这个单位就是千克 × 厘米，我们还很难说这个单位有什么意义。

习题9-1

请编写一个计算两个数据序列协方差的函数 `Cov`，为了测试你写的函数，可以计算两个相同序列的协方差，确保有 $\mathrm{Cov}(X, X)=\mathrm{Var}(X)$。

读者可以从 http://thinkstats.com/correlation.py 下载到答案。

9.3 相关性

解决上一节协方差遇到的问题的方法是用标准分数来代替原始的值，计算两个标准分数的乘积：

$$p_i = \frac{(x_i - \mu_X)}{\sigma_X}\frac{(y_i - \mu_Y)}{\sigma_Y}$$

这些乘积的均值为：

$$\rho = \frac{1}{n}\sum p_i$$

这个值称为皮尔逊相关系数（Pearson’s correlation），用以纪念现代统计学创立者卡尔·皮尔逊。相比于协方差，相关系数很容易计算，更重要的是，它的结果更容易解释。相关系数的单位为 1。

相关系数 ρ 的取值为 −1 到 1 之间。我们改写一下 ρ 的表达形式就可以很容易得到这个结果：

$$\rho = \frac{\mathrm{Cov}(X,Y)}{\sigma_X\sigma_Y}$$

将离差项代入公式，可得

$$\rho = \frac{\sum \mathrm{d}x_i \mathrm{d}y_x}{\sum \mathrm{d}x_i \sum \mathrm{d}y_i}$$

利用著名的柯西—施瓦兹不等式[1]（Cauchy-Schwarz inequality）即可证明 $\rho^2 \leqslant 1$，故而有 $-1 \leqslant \rho \leqslant 1$。

ρ 的绝对值的大小代表两个变量相关的程度。当 ρ=1 时，两个变量完全相关，即如果我们知道了其中一个变量的值，就可以精确预测另一个变量的值。ρ=−1 时也是同样的情况，只是两个变量是完全负相关而已。

现实中大部分的相关都没有这么完全，但是相关系数仍然提供了一些有用的信息。例如，在知道了一个人的身高后，我们猜测这个人的体重，虽然我们不大可能猜对，但是相比于不知道身高的情况下，我们依然可以猜测得更准确。皮尔逊相关系数衡量了我们能够多准确地猜测结果。

注 1：参见 http://wikipedia.org/wiki/Cauchy-Schwarz_inequality。

如果 ρ=0，这是不是意味着两个变量之间毫无关系呢？不幸的是我们不能得出这个结论。皮尔逊相关系数只是衡量两个变量之间的线性关系。如果两个变量之间的关系不是线性的，那么 ρ 可能低估两个变量之间的相关性。

图 9-1 来源于 http://wikipedia.org/wiki/Correlation_and_dependence。图中展示了一些精心构造的数据的散点图及对应的相关系数。

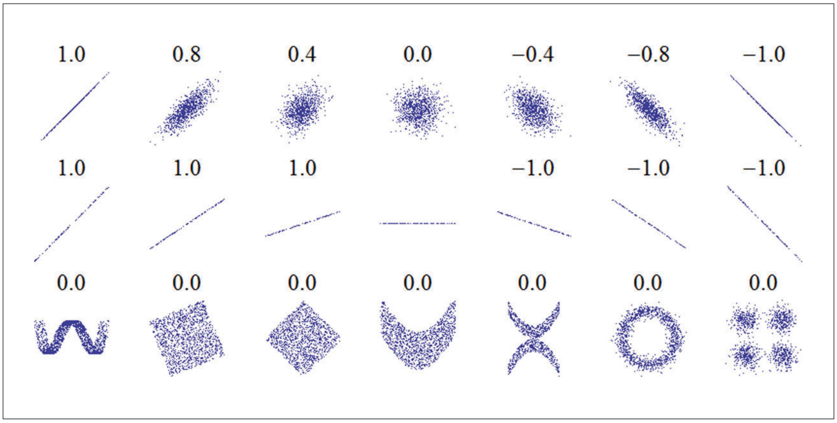


图 9-1：具有一定相关性的示例数据集

图中第一行展示了几组有线性相关性的数据的相关系数，我们可以直观地了解这些相关系数大约会对应到什么水平的关系。第二行展示了完全相关的数据的相关系数，这里我们发现相关系数跟斜率是无关的（稍后就会讲到估计斜率）。第三行展示了一些有明显相关性的数据，但由于这些关系不是线性的，这里的皮尔逊相关系数等于 0。

这提醒我们别盲目地相信这个系数，在计算相关系数之前，一定要画个散点图观察一下数据。

习题9-2

请编写一个计算相关系数的函数 `Corr`，该函数可接受两组数据。提示：这里可以用到之前的函数 `thinkstats.Var` 和 `Cov`。

可以通过计算 Corr(*X*, *X*) 是否等于 1 来测试函数是否正确。http://thinkstats.com/correlation.py 提供了一个答案。

9.4 用pyplot画散点图

散点图是探测两个变量是否具有相关性的最简单的方法，但是要画一张高质量的散点图并不容易。接下来我们用 4.5 节所提到的行为危险因素监控系统（BRFSS）中的身高和体重做图。`pyplot` 中提供了一个画散点图的函数 `scatter`：

```
import matplotlib.pyplot as pyplot
pyplot.scatter(heights, weights)
```

图 9-2 是画出来的结果。从图中可以看到二者确实是正相关的，身高较高的人体重较大。但这个结果并不理想，图中明显可以看到数据点被分成一列列的。这是因为在收集身高数据时，是以英寸为基本单位收集的，转换成厘米之后也进行了取整。这样就丢失了一些信息。

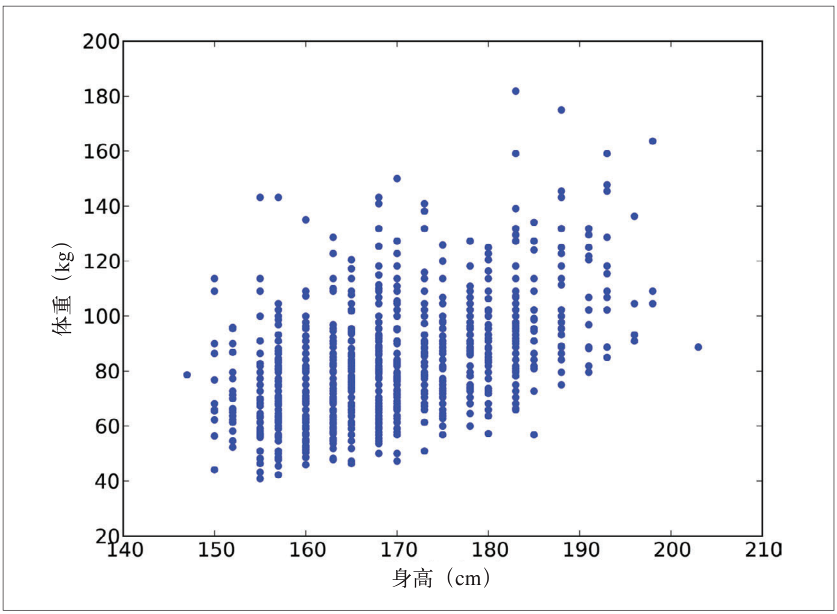


图 9-2：BRFSS 中被调查者体重—身高简易散点图

显然那些丢失的信息已经无可挽回，但我们可以给每个身高数据加一个随机扰动来尽可能地使数据回到之前的样子。因为数据是以英寸为单位收集的，我们可以给它们加上一个扰动（jitter），这里给它们加上一个 [−0.5, 0.5] 英寸上的均匀分布的随机数，换算成厘米后的范围是 [−1.3, 1.3]：

```
jitter = 1.3
heights = [h + random.uniform(-jitter, jitter) for h in
heights]
```

图 9-3 展示了处理后的数据的散点图。图中的趋势更加明显了。一般情况下我们可以通过这种方法来得到更有说服力的图，不过，在进行数据分析时就必须用原始数据。

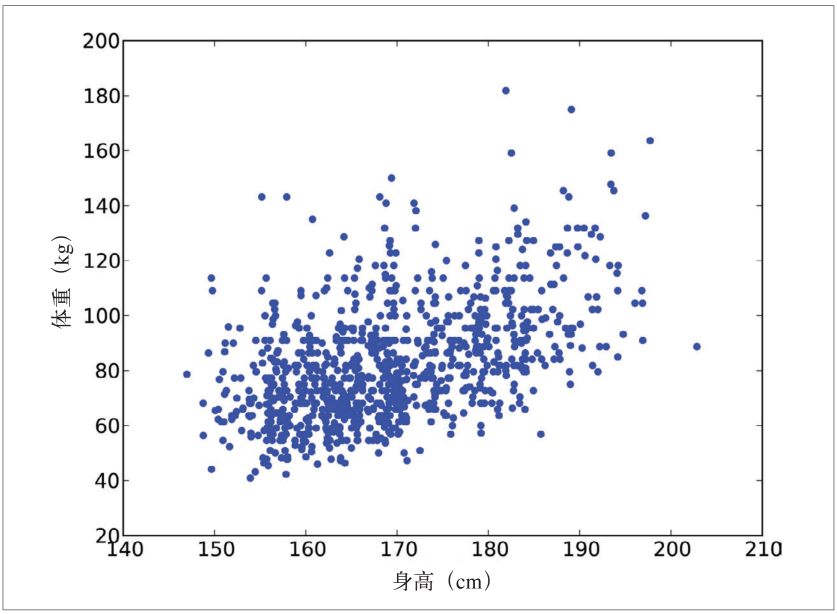


图 9-3：数据进行扰动处理后的散点图

即使进行了上述数据扰动处理，我们可能依然无法很好地展示数据。因为图中有很多点重叠在一起，这样就隐藏了图中密集度高的一些数据的信息，同时可能会过分凸显那些异常值。

我们可以引入一个透明度参数 α 来解决这个问题：

```
pyplot.scatter(heights, weights, alpha=0.2)
```

图 9-4 展示了增加透明度后的结果。图中有重叠的点看起来颜色更深了，这样颜色深度就跟点的密度成比例地变化。图中在 90 kg 附近有一个明显的横线，数据是被调查者提供的，所以这里最可能的原因是人们对体重的数据进行了取整（或许他们想让自己的体重看起来更轻一些）。

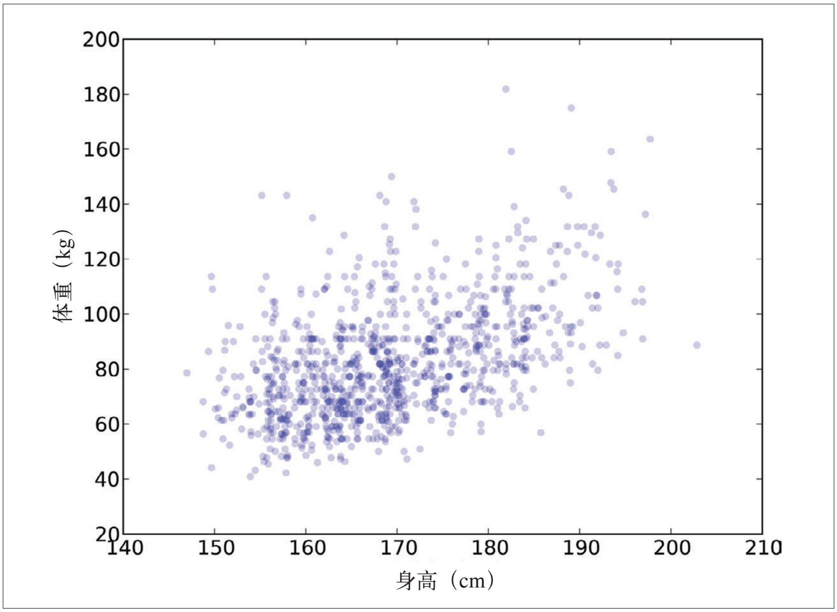


图 9-4：数据进行扰动处理并引入透明度参数后的散点图

这样的图很适合用在那些数据量不是很大的情况。这里我们仅用了 BRFSS 中的 1000 个数据，而 BRFSS 总共包含了 414 509 个数据。

当我们要处理大量的数据时，上述方法可能看起来都会一团糟。我们可以用一种称为 `hexbin` 的方法来处理这样的问题：首先将图分成一个个小格子，统计每个格子中有多少个数据点，然后根据格子中点的个数来上色。`pyplot` 提供了一个 `hexbin` 函数来实现这个功能：

```
pyplot.scatter(heights, weights, cmap=matplotlib.cm.Blues)
```

图 9-5 展示了 hexbin 的结果。这种图的好处是能展示出数据关系的整体形状，对于大型数据集非常高效，但缺点是我们可能看不见那些异常值了。

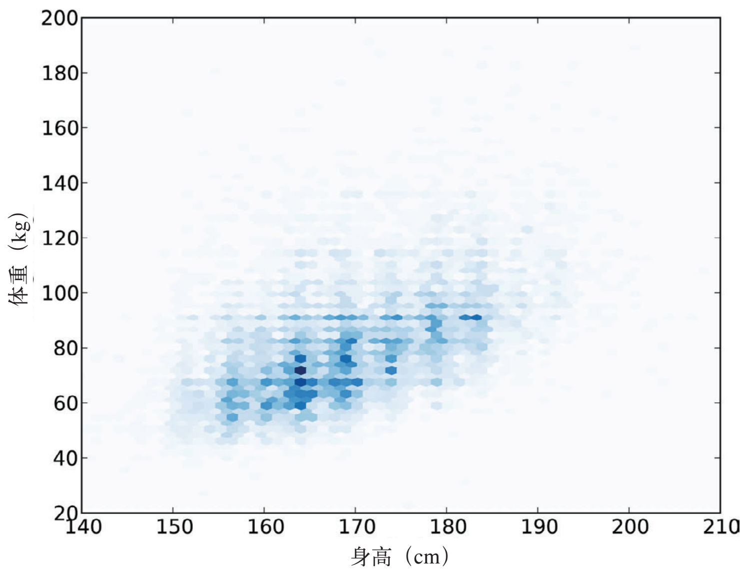


图 9-5：使用 **hexbin** 函数绘制的散点图

上述例子告诉我们，要画一张能真实反映数据关系的散点图并不是一件容易的事情。读者可以从 http://thinkstats.com/brfss_scatter.py 下载到本节画图所用的代码。

9.5 斯皮尔曼秩相关

如果两组数据的变量分别大致正常，而且两者呈线性关系，那么皮尔逊相关系数可以很好地刻画它们之间的关系。但是皮尔逊相关系数对异常值的影响很敏感。

Anscombe 构造的 4 组数据（Anscombe’s quartet）很明显地说明了这个问题。这 4 个数据集有相同的相关系数：第一个是两组变量呈线性关系，但有随机噪声的影响；第二个是两组变量呈某种非线性的函数关系；第三个是两组变量呈完全的线性关系，但是有一个异常值；最后一个是两组变量并无相关性，但一个异常值除外。读者可以从 http://wikipedia.org/wiki/Anscombe’s_quartet 了解到更具体的信息。

斯皮尔曼秩相关系数（Spearman’s Rank Correlation）可以用在存在异常值和变量分布非常不对称的情况。为了计算斯皮尔曼秩相关系数，我们先计算序列中数值的*秩*（rank），即某个值在序列中按大小排序后的位置。例如在序列 {7, 1, 2, 5} 中，值 5 的秩等于 3，因为按从小到大排序，5 在这个序列中排第 3 位。将序列转换成秩之后，再计算皮尔逊相关系数，得到的结果就是斯皮尔曼秩相关系数。

除了斯皮尔曼秩相关系数，另一种方法是对原始的数据做一个变换，使得变换之后的结果接近正态分布，然后再算皮尔逊相关系数。例如，如果数据近似服从对数正态分布，那么我们可以先对数据取对数，然后再算相关系数。

习题9-3

请编写一个计算数据序列的秩的函数。例如，设一个数据序列为 {7, 1, 2, 5}，做秩转换后结果为 {4, 1, 2, 3}。

如果数据序列中出现多个相同的值，那么严格的做法是给这些数值赋予一个秩的平均值。但如果不这么严格，而是随意地给这些值安排个顺序，一般也不会造成什么误差。

请编写一个函数，计算两个数据序列的斯皮尔曼秩相关系数。读者可以从 http://thinkstats.com/correlation.py 下载本题的答案。

习题9-4

请下载 http://thinkstats.com/brfss.py 和 http://thinkstats.com/brfss_scatter.py 并运行这些代码。确保你能读懂 BFRSS 数据，然后生成散点图。

将生成的图与图 9-1 做比较，你期望这里的皮尔逊相关系数会是多少？计算得到的结果呢？

成人体重大致服从对数正态分布，也有相关的异常值影响。请绘制体重的对数与身高的散点图，并计算变换后的皮尔逊相关系数。

最后，请计算体重和身高的斯皮尔曼秩相关系数。你觉得哪个相关系数能更好地描述这两个变量关系的强度？可从 http://thinkstats.com/brfss_corr.py 下载到问题的答案。

9.6 最小二乘拟合

相关系数可以衡量两个变量之间线性相关的强度和正负，但是无法知道它们的斜率。有很多方法可以用来估计斜率，其中线性*最小二乘拟合*（linear least square fit）是最常用的一种方法。线性拟合（linear fit）指的是用一个线性的方程来拟合两个变量之间的关系。最小二乘法（least square）是使拟合函数与数据之间的均方误差达到最小的拟合方法[2]。

假设我们有一个数据序列 X，要通过 X 的一个函数来预测另一个数据序列 Y。如果这个预测函数是线性的，截距为 α，斜率为 β，那么我们可以预期 y_i 大约会等于 $\alpha+\beta x_i$。

除非这两个序列是完全线性相关的，否则我们只能近似地预测 Y 的值。预测的离差 (或称*残差*) 为：

$$\varepsilon_i = (\alpha + \beta x_i) - y_i$$

残差的出现可能是由数据测量误差造成的，也可能是一些我们未知的非随机因素引起的。例如，当我们通过身高的一个函数来预测体重时，这些未知的因素可能就包括饮食、身体锻炼情况和体型等。

假设两个变量存在这样的一个线性关系，那么如果我们错误地估计了

注 2：参见 http://wikipedia.org/wiki/Simple_linear_regression。

参数 α 和 β，就会造成很大的残差。所以，这些参数自然应该使得残差尽可能地小。

一般地，可以通过最小化残差的绝对值、平方、立方等来求解参数。实际中最通用的方法是使残差的平方和最小，即

$$\min_{\alpha,\beta}\sum \varepsilon_i^2$$

我们解释一下这个选择的原因。

- 平方能将正残差和负残差都变成正数，这符合我们的目标。
- 平方相当于给残差赋予了一个权重，越大的残差（绝对量）被赋予的权重越大。但是并不是所有情况下大的残差都应该被赋予大的权重，因为这样拟合方程就很容易受到异常值的影响。
- 在残差服从均值为 0、方差为 σ^2（未知，但为常数）的正态分布，且在残差与 x 独立的假设下，参数的最小二乘估计结果与极大似然估计量相同。[3]
- 最小二乘估计的计算非常简单。

就现在而言，除非在某些计算效率比方法更重要的情况下，否则上述最后一个原因已经不再那么具有吸引力了。所以更多时候要思考的是，就我们的问题而言最小二乘是否是最适合的方法。

例如，我们用 X 来预测 Y。如果预测结果偏高造成的代价远低于预测结果偏低造成的代价，那么构造一个损失函数 $\text{cost}(\varepsilon_i)$，并最小化损失函数会是一个更好的选择。

接下来我们介绍如何进行最小二乘拟合。

1. 计算两个序列的均值 $\bar{x}$ 和 $\bar{y}$，X 的方差，X 和 Y 的协方差。

2. 估计斜率

$$\hat{\beta} = \frac{\text{Cov}(X,Y)}{\text{Var}(X)}$$

注 3：请参考 Press 等人合著的 *Numerical Recipes in C* 第 15 章：http://t.cn/zYeSWUm。

3. 估计截距

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

读者可以从 http://wikipedia.org/wiki/Numerical_methods_for_linear_least_squares 了解公式的推导过程。

习题9-5

请编写一个名为 `LeastSquares` 的函数，用来估计 X 和 Y 的回归系数 $\hat{\alpha}$ 和 $\hat{\beta}$。可以从 http://thinkstats.com/correlation.py 下载到问题的答案。

习题9-6

在 BRFSS 的数据中，请用身高对体重的对数进行最小二乘拟合。可以从 http://thinkstats.com/brfss_corr.py 下载到问题的答案。

习题9-7

某个地点风速的分布决定了该地点的风能密度。风能密度是安装在该位置的风力涡轮机所能产生的平均功率的上限。根据之前的一些研究，风速的经验分布非常接近威布尔分布（参考 http://wikipedia.org/wiki/Wind_power#Distribution_of_wind_speed）。

为了评估某个地方是否有安装风力涡轮机的价值，我们可以在这个地方设置一个风力计来测量一段时间内的风速。但很难测量到风速分布的尾巴，因为可以认为这是一个小概率事情，一般不大可能在一次试验中被观测到。

一种解决这个问题的方法是，我们先估计威布尔分布的参数，再对分布求积分计算风能密度。

为了估计威布尔分布的参数，可以用习题 4-6 中的变换方法，然后用最小二乘拟合来计算变换数据的斜率和截距。

请编写一个从威布尔分布中抽取样本并估计分布参数的函数。

最后请编写一个利用威布尔分布参数计算平均风能密度的函数。（这里你可能需要了解一下风力方面的专业知识。）

9.7 拟合优度

在用线性模型拟合完数据之后，我们需要评估模型拟合的好坏情况。当然，这种评估取决于我们想要用这个模型来做什么。一种评估模型的办法是计算模型的预测能力。

在一个预测模型中，我们要预测的值称为*因变量*（dependent variable），而用于预测的值称为*解释变量*或*自变量*（explanatory variable 或 independent variable）。

我们可以通过计算模型的*确定系数*（coefficient of determination），也即通常所说的 R^2，来评价模型的预测能力：

$$R^2 = 1 - \frac{\mathrm{Var}(\varepsilon)}{\mathrm{Var}(Y)}$$

我们通过一个例子来解释一下 R^2 的意义。假设你试图去猜测一群人的体重是多少，你知道这群人的平均体重是 $\bar{y}$。如果除此之外你对这些人一点儿都不了解，那么你最佳的策略是选择猜测他们所有人的体重都是 $\bar{y}$。这时，估计的均方误差就是这个群体的方差 Var(Y)：

$$\mathrm{MSE} = \frac{1}{n}\sum(\bar{y} - y_i)^2 = \mathrm{Var}(Y)$$

接下来，假如我告诉你这群人的身高信息，那么你就可以猜测体重大约为 $\hat{\alpha} + \hat{\beta}x_i$，在这种情况下，估计的均方误差就为 Var($\varepsilon$)：

$$\mathrm{MSE} = \frac{1}{n}\sum(\hat{\alpha} + \hat{\beta}x_i - y_i)^2 = \mathrm{Var}(\varepsilon)$$

所以，Var(ε)/Var(Y) 表示的是有解释变量情况下的均方误差与没有解释变量情况下的均方误差的比值，也即不能被模型解释的均方误差占总的均方误差的比例。这样 R^2 表示的就是能被模型解释的变异性的比例。

假如一个模型的 R^2=0.64，那么我们就可以说这个模型解释了 64% 的变异性，或者可以更精确地说，这个模型使你预测的均方误差降低了 64%。

在线性最小二乘模型中，我们可以证明确定系数和两个变量的皮尔逊相关系数存在一个非常简单的关系，即：

$$R^2 = \rho^2$$

具体证明可参考 http://en.wikipedia.org/wiki/Coefficient_of_determination。

习题9-8

韦克斯勒成人智力量测验（Wechsler Adult Intelligence Scale，WAIS）是一种测量智商的方法。测量的分数都进行了校正，使得其人群中的均值为 100，标准差为 15。

假设你想通过一群人的 SAT 成绩来预测这些人的 WAIS 分数。根据之前的一项研究，SAT 成绩与 WAIS 分数之间的皮尔逊相关系数为 0.72。

如果将你的模型用到一个非常大的人群中，你预期预测的均方误差会是多少？

提示：如果总是预测 WAIS 分数等于 100，那么均方误差会是多少呢？

习题9-9

编写一个名为 `Residuals` 的函数，根据 X、Y、$\hat{\alpha}$ 和 $\hat{\beta}$，计算残差 ε_i。

编写一个名为 `CoefDetermination` 的函数，根据 ε_i 和 Y，计算 R^2。要测试函数的正确性，看是否有 R^2=ρ^2。读者可以从 http://thinkstats.com/correlation.py 下载问题的答案。

习题9-10

根据 BRFSS 中身高和体重的数据，计算 $\hat{\alpha}$ 、$\hat{\beta}$ 和 R^2。如果需要预测一

个人的体重，那么他的身高会对你起多大的帮助？读者可以从 http://thinkstats.com/brfss_corr.py 下载问题的答案。

9.8 相关性和因果关系

一般说来，两个变量之间的相关关系并不能告诉我们一个变量的变化是否是由另一个变量的变化引起的；或许两个变量就没有直接的因果关系，而是因为其他原因导致二者同时发生了相同（或相反）趋势的变化；也或许就是因为偶然因素造成了它们之间的相关性。（参考 http://xkcd.com/552/。）

维基百科的“Correlation does not imply causation”简要列出了可能导致两个变量产生相关性的原 因。读者可以参考 http://en.wikipedia.org/wiki/Correlation_does_not_imply_causation。

那么我们怎么样才能从相关性的信号中得到因果关系的结果呢？

1. 利用时间的先后关系。如果 A 事件在 B 事件之前发生，那么 A 就有可能是导致 B 的原因（但反之不成立，根据因果关系的常识是这样的）。事件发生的顺序可以帮助我们推断因果顺序，但是这并不能排除存在另外一些事件导致了 A 和 B 的发生。

2. 利用随机性。如果我们将一个非常大的总体随机分成两部分，然后分别计算这些变量在两个子总体中的平均值。我们可以期望这些均值的差异会很小，因为中心极限定理保证了这一结果。
   如果有一个变量在两个分组中有明显的差别，而其他所有的变量在两个分组里面都几乎一样，这时我们就可以排除掉一些虚假的相关性。即使相关变量未知，这个方法也行得通，当然知道会更好，这样我们就能检查两个分组是否一致。

随机对照试验（randomized controlled trial）就是根据这些想法设置的。在这种试验中，被试者被随机地分成两组（或多组）：实验组（treatment group）会接受某种干预，例如服用某种新药；而对照组（control group）则不接受这种干预，或者只接受已知效应的处理。

随机对照试验的结果在因果关系的鉴定上是最可信赖的方法之一，在循证医学中有广泛的应用。参考 http://wikipedia.org/wiki/Randomized_controlled_trial。

不幸的是，随机对照试验只能用于实验研究、药物研发等少数情况。社会科学很少用到这种方法，因为这种试验可能无法进行，或者会引起伦理争端。

另外一种方式是进行*自然试验*（natural experiment）。在这种试验中，我们尽量控制群体在各个方面都是相似的，然后对不同的群体实施不同的处理。这里会涉及的一个问题是各个群体可能存在一些我们观测不到的差异。维基百科上给出了更详细的信息 http://wikipedia.org/wiki/Natural_experiment。

在某些情况下我们能通过*回归分析*（regression analysis）推断出因果关系。线性最小二乘是用自变量解释因变量的简单回归。本章只讨论了有一个自变量时的情况，多个自变量的回归所用的技术与本章所用的基本相同。

本书并未涉及这些技术，有很多种方法可以控制虚假的相关性。例如，在 NSFG 的数据中，我们发现第一胎婴儿的体重倾向于比非第一胎婴儿的轻（参见 3.6 节），但是婴儿出生的体重还与母亲的年龄有关。而生第一胎婴儿的母亲的年龄倾向于比生非一胎婴儿母亲的小。

所以有可能是生第一胎婴儿的母亲更年轻导致了第一胎婴儿的体重更轻。为了控制年龄的影响，我们可以将母亲按年龄大小分组，然后比较同一个分组中第一胎婴儿的体重和非第一胎婴儿的体重。

如果这时体重的差异依然存在，那么我们就可以说这种体重上的差异跟母亲的年龄无关。但如果这时各个分组之间的体重差异消失了，那么我们得到的结论就是这种体重上的差异完全是由母亲的年龄造成的。或者，如果这些体重差异变小了，我们可以计算出母亲年龄对体重差异造成了多大的影响。

习题9-11

NSFG 的数据中包含了一个记录婴儿出生时母亲年龄的变量 `agepreg`。绘制母亲年龄和婴儿出生体重的散点图，两者之间关系如何？

用母亲年龄对婴儿出生体重进行最小二乘拟合。估计量 $\hat{\alpha}$、$\hat{\beta}$ 的单位是什么？是否可以用一两句话描述这个拟合结果的意义。

计算生头胎婴儿的母亲年龄的均值和生非头胎婴儿的母亲年龄的均值。两个分组的母亲年龄均值差异有多大才会影响到婴儿出生体重的差异？婴儿出生体重的差异有多大比例可以用分娩时母亲年龄的差异来解释？

读者可以从 http://thinkstats.com/agemodel.py 下载问题的答案。如果读者对多元回归感兴趣，可以浏览 http://thinkstats.com/age_lm.py，这里展示了如何通过 Python 调用 R 这个统计计算软件。

9.9 术语

- 确定系数（coefficient of determination）
  衡量模型拟合结果好坏的指标。

- 对照组（control group）
  对照试验中没有接受处理的组，或受到已知效应处理的组。

- 相关性（correlation）
  对两个变量关系的一种描述。

- 协方差（covariance）
  衡量两个变量变化方向是否一致的统计量。

- 因变量（dependent variable）
  我们想要解释或者预测的变量。

- 自变量（independent variable）
  用于预测因变量的变量，也称解释变量。

- 最小二乘拟合（least squares fit）
  最小化残差平方和的数据拟合方法。

- 自然试验（natural experiment）
  一种试验设计的方法，就是利用自然形成的界限将受试者分成几个分组，并且大体上使得分组结果接近随机分组。

- 归一化（normalize）
  将一组数据进行转换，使其均值为 0，方差为 1。

- 随机对照试验（randomized controlled trial）
  一种试验设计的方法，将受试者随机分成几个分组，并对不同的分组实施不同的处理。

- 秩（rank）
  将一个序列按大小排序后，序列中的某个元素所处的位置。

- 残差（residual）
  衡量模型预测结果与真实值离差的值。

- 标准分数（standard score）
  归一化后的值。

- 处理（treatment）
  对照试验中对一个分组所做的干预或改变。

作者及封面简介

关于作者

Allen Downey 是富兰克林欧林工程学院的计算机科学副教授，曾执教于韦尔斯利学院、科尔比学院和加州大学伯克利分校。他先后获得麻省理工学院计算机科学学士和硕士学位，加州大学伯克利分校计算机科学博士学位。

关于封面

本书封面上的动物是喷水鱼，封面图片来自 *Dover*。

索引

T

W

X

Y

Z

Statistics Programming

现实工作中，人们常被要求用数据说话。可是，数据自己是不能说话的，只有对它进行可靠分析和深入挖掘才能找到有价值的信息。概率统计是数据分析的通用语言，是大数据时代预测未来的根基。

站在时代浪尖上的程序员只有具备统计思维才能掌握数据分析的必杀技。本书正是一本概率统计方面的入门图书，但视角极为独特，折射出大数据浪潮的别样风景。作者将基本的概率统计知识融入Python编程，告诉你如何借助编写程序，用计算而非数学的方式实现统计分析。一个趣味实例贯穿全书，生动地讲解了数据分析的全过程：从采集数据和生成统计量，到识别模式和检验假设。一册在手，让你轻松掌握分布、概率论、可视化以及其他工具和概念。

- 编写测试代码深入理解概率统计
- 运行实验检验统计行为特征
- 通过模拟理解数学上艰涩的概念
- 学习贝叶斯估计等实用内容
- 用Python导入各种来源的数据
- 运用统计推断解决真实数据问题

Allen B. Downey是富兰克林欧林工程学院计算机科学副教授，加州大学伯克利分校计算机科学博士。Downey已出版十余本技术书，内容涉及Java、Python、C++、概率统计等，深受专业读者喜爱。他的最新Think系列书还有*Think Complexity: Complexity Science and Computational Modeling*、*Think Python*。

O'REILLY®
oreilly.com.cn

封面设计：Karen Montgomery　张健
图灵社区：www.ituring.com.cn
新浪微博：@图灵教育 @图灵社区
反馈/投稿/推荐信箱：contact@turingbook.com
热线：(010)51095186转604

分类建议　计算机/计算机数学

人民邮电出版社网址：www.ptpress.com.cn
O'Reilly Media, Inc.授权人民邮电出版社出版

ISBN 978-7-115-31737-7
9 787115 317377 >

ISBN 978-7-115-31737-7
定价：29.00元